数理解析研究所講究録 1286 巻 2002 年 155-169

A recognition method of matrices by using variable block pattern elements generating rectangular area

Toshihiro Kanahori

Tsukuba College of Technology, Kasuga 4–12–7, Tsukuba, 812–8581 Japan
kanahori@k.tsukuba-tech.ac.jp

Abstract. In this paper, we propose our new method to recognize matrices including *repeat symbols* and *area symbols*. The method consists of 4 parts; detection of matrices, segmentation of elements, construction of networks and analysis of the matrix structure. In the construction of networks, we regard a matrix as a network of elements connected each other by links representing their relative relations, and consider its horizontally projected network and vertically projected one. In the analysis, we obtain the areas of variable block pattern elements generating the minimum rectangular area of the matrix by solving the simultaneous system of equations given by the two projected networks. We also propose a format to represent the structure of matrices to output the result of the matrix recognition.

1 Introduction

The technology of OCR is very efficient to digitize printed documents. However, current OCR systems can not recognize mathematical formulae which are very important in scientific documents. Several algorithms for recognizing mathematical formulae have been reported in literature ([1]-[3]). Some of them can be applied to very simple matrices, such as gridironed matrices. However, no method to recognize matrices including abbreviation symbols, which are used in mathematics, is reported. Besides, there is no standard format to represent the structure of complicated matrices. So, we can not keep the result of matrix recognition.

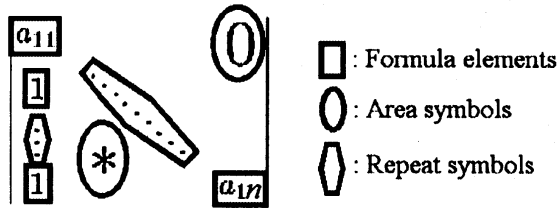


Fig. 1. Components of matrix

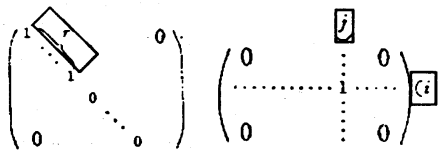


Fig. 2. Decorators

In this paper, we present a method to recognize matrices including *repeat symbols* or *area symbols*, which appear in scientific documents, and a format

to represent their structure to output the result of our recognition, and report the experimental results of this method. Matrices which we are going to recognize consist of *formula elements*, *area symbols* and *repeat symbols* (Fig. 1). The *decorators* are excluded at present (Fig. 2).

2 Representation of matrix

We classify the components of a matrix into the following 3 classes;

1. Formula element
   - It is a component of a matrix.
   - It has only one grid as its own area.
   - It can connect to other elements in the 8 directions.
2. Area symbol
   - It has several grids as its own area.
   - Its area has a free boundary.
   - Common area symbols are $O, 0, 1, *$, etc., and a space is also an area symbol.
3. Repeat symbol
   - It means that formula elements are continuously aligned on the straight line in its direction; $\downarrow, \rightarrow, \searrow$ or $\swarrow$.
   - It can connect to formula elements and other repeat symbols with different directions.
   - We assume that it consists of 3 points or more and they are put on straight line.

We represent an area of a matrix element by the set of couples of indices representing the row and column on the matrix. In Fig. 3, the formula element 'a_{11}' has (1,1) as its area, 'a_{nn}' has (4,4), and the area symbol '0' has (1,2), (1,3), (1,4), (2,3), (2,4) and (3,4) as its area, and '$*$' has (3,2), (4,2), (4,3).

The format to represent the structure of matrices is resumed in Table 1. The results of our matrix recognition are output in it. For example, the matrix in Fig. 3 is represented by Table 2, where we omitted the coordinates of the elements' bounding rectangles.

Symbol Names	Informations
MATRIX	Coordinate of its own bounding rectangle on the image
	Parentheses on right and left
	Numbers of the row and column
	List of ELEMENT
	List of CONNECTION
ELEMENT	Coordinate of its bounding rectangle
	Results of the recognition
	Set of its areas
CONNECTION	Couple of the positions of the repeat symbol's end points

Table 1. The rule of matrix representation

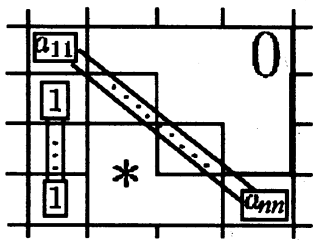


Fig. 3. Positions on matrix

MATRIX			
Parentheses	|, |	Row, Column	4, 4
ELEMENT LIST			
Formula	Areas	Formula	Areas
a_{11}	(1,1)	1	(2,1)
1	(4,1)	a_{nn}	(4,4)
0	(1,2),(1,3),(1,4),(2,3),(2,4),(3,4)		
*	(3,2),(4,2),(4,3)		
CONNECTION LIST			
End Points	(2,1),(4,1)	End Points	(1,1),(4,4)

Table 2. Example of matrix representation

3 Matrix recognition

For our matrix recognition of a page image, we assume that its lines are distinguished, the characters are recognized, the coordinate of the bounding rectangle of each character is obtained.

The method consists of 4 parts;

1. Detection of matrices in a page image, and extraction of characters in each area of the matrices.
2. Segmentation of the characters into elements for each matrix.
3. Construction of the network where formula elements are connected by repeat symbols or adjacent relations.
4. Structure analysis of the matrix.

In the 4th step, we let the minimum length of repeat symbols in a matrix be 2 on the network. Then, we set up equations for the height and width of the matrix from its vertically projected network and horizontally one. By solving the equations, we obtain the areas of variable block pattern elements generating the minimum rectangular area of the matrix, and decide the minimum numbers of its rows and columns.

3.1 Detection of matrices

The algorithm of the detection of matrices is very simple at present. Its outline proceeds as follows;

1. Find big parentheses in the given character sequence. Considering errors of the character recognition, find long tall characters too.
2. Find couples of big parentheses among them.
3. For each couple, calculate the rectangular area between its two parentheses, and recognize characters in the area of the matrix.

In the followings, we assume that the results of the character recognition are always correct.

3.2 Segmentation of elements

By the detection of matrices, each detected matrix has the set of characters in its own area. It is necessary to group them into matrix elements. In this section, we explain the method of the segmentation.

We let $L = \{C_1, \cdots, C_n\}$ be the set of the characters in the matrix. We define the distance $d(C, D)$ between C and $D \in L$ by

$$d(C, D) := \alpha_x d_x(C, D) + \alpha_y d_y(C, D),$$

where d_x (or d_y) is the distance between the intervals projected to x-axis (resp. y-axis), but we let d_x (resp. d_x) be 0 if the intersection of the intervals is not empty. The coefficients, α_x and α_y, also depend on C and D. We let the coefficient α_x for the horizontal distance be smaller than α_y for the vertical distance, so that the horizontal connections are tighter than the vertical ones. If C or D is a binary operator, we set α_x smaller value than ordinary. The operator, '$-$' (minus), is also used as a sign at head of elements. Therefore, if the left space of a character '$-$' (minus) is longer than the right, we consider it as a sign, and cut the connection to its left-hand side. Big symbols, $\sum$, $\prod$, etc., and fractional lines often have vertical connections. If they have formulae above themselves or below (upper limit formulae, lower one, numerators or denominators), the formulae are closer to them. So, we let α_y smaller to prevent them from connecting to the other elements which are above or below.

According to the way of the segmentation, it is not necessary to calculate the distance between a pair of characters where they are clearly unadjacent each other (In Fig. 4, 'a' is unadjacent to 'c' and 'O').

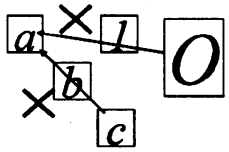


Fig. 4. Unadjacent

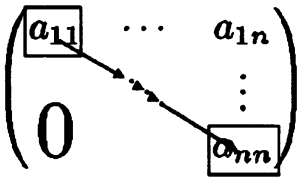


Fig. 5. Extraction of repeat symbols

However, the distance between clearly unadjacent characters is useful for evaluation of the thresholds. For a character C, we define the set of clearly unadjacent distance, $T(C)$, by

$$T(C) := \{d_0(C, D) | D : \text{clearly unadjacent to } C\}$$
$$(\ d_0 := d_x(C, D) + d_y(C, D)\ ).$$

We put the threshold with respect to C by

$$t(C) = \frac{d(C, D_1) + d(C, D_2)}{2},$$

where D_1 (or D_2) is the character whose clearly unadjacent distance to C, $d_0(C, D_1)$ (resp. $d_0(C, D_2)$), is the first (resp. second) minimum value in $T(C)$.

We let $G(L)$ be the directed graph derived from the adjacency matrix $A(L) := (a_{ij})_{i,j=1,\cdots,n}$. Then, we can obtain the elements of the matrix as the connected

components of the graph $G(L)$, and recognize each element again, where the elements a_{ij} of $A(L)$ are defined by

$$a_{ij} := \begin{cases} 1 \ (d(C_i, C_j) < t(C_i)) \\ 0 \ (d(C_i, C_j) \geq t(C_i)) \end{cases} \quad (C_i, C_j \in L).$$

The special process for the class of dots is done after the above segmentation. First, we take a dot which have very close elements on its both sides for a comma, and combine them into one element. Next, we extract repeat symbols from the dots. We classify repeat symbols into 4 types ($\downarrow, \rightarrow, \searrow, \swarrow$) according to their directions. The extraction of repeat symbols proceeds by tracing the dots (Fig. 5).

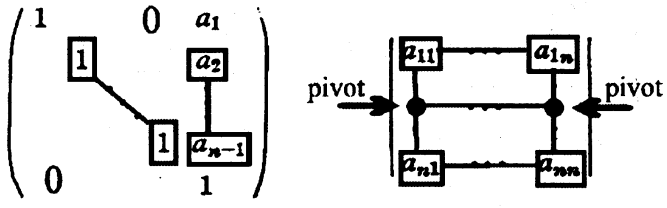
3.3 Construction of networks and equations

This section describes the algorithm to construct *the horizontally projected network* and *the vertically projected network* using the following simple examples.

$$\begin{pmatrix} 1 & & 0 & a_1 \\ & 1 & & a_2 \\ & & \ddots & \vdots \\ & & 1 & a_{n-1} \\ 0 & & & 1 \end{pmatrix} \qquad \begin{vmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & \cdots & \vdots \\ a_{n1} & \cdots & a_{nn} \end{vmatrix}$$

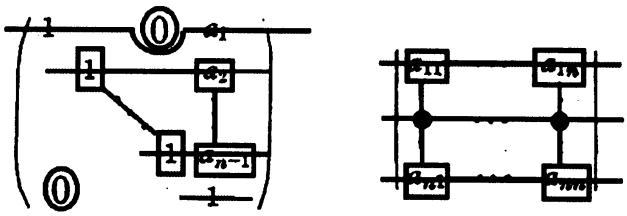
1. Connection by repeat symbols

For each repeat symbol, we connect its origin to its terminal. If the origin and the terminal are other repeat symbols, we put pivots on them. We consider the lengths of these connections as variable. If a repeat symbol is divided by a pivot, let the minimum value of the variable corresponding to the symbol be 1. Otherwise, let the minimum values of repeat symbols be 2.



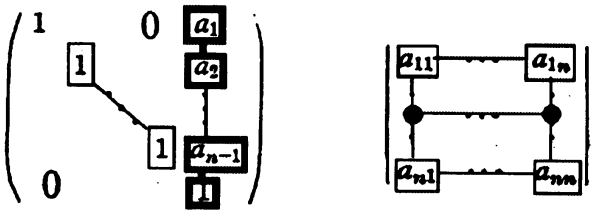
2. Segmentation into lines

First, we set each pair of the horizontally connected elements on the same line. Second, we segment the elements into lines by using the lengths of overlapping of their bounding rectangles on their horizontal projection, their sizes and baselines. If there are bigger $O, 0, 1, *$ than their normal sizes, or elements laying on several lines, we let them be area symbols.



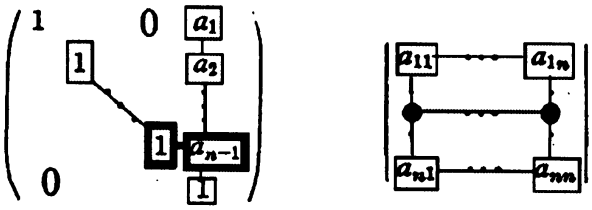
3. Vertical connection of elements

We connect each pair of vertically adjacent elements by a vertical 1-length path, its vertical length is 1, and its horizontal length is 0.



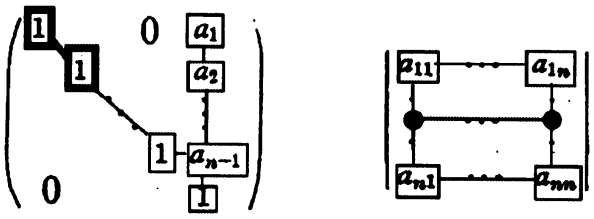
4. Horizontally connection of elements

We connect each pair of horizontally adjacent elements by a horizontal 1-length path, its vertical length is 0, and its horizontal length is 1.



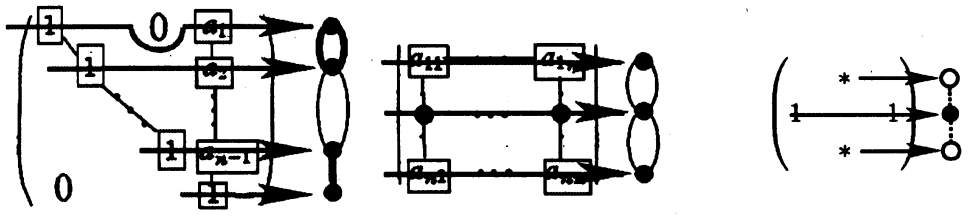
5. Diagonal connection of elements

For each element which is not connected to others by 1-length paths, we connect it to its diagonally adjacent element by a diagonal 1-length path, its vertical length is 1, and its horizontal length is 1.



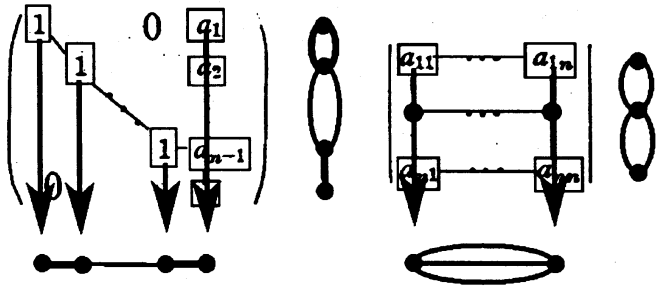
6. Horizontal projection of the network

By identifying elements on each line, we horizontally project the network constructed on the matrix by the above connections. We also identify elements on the upper end (or the lower end) of the matrix. If the projected area covers some area symbols, we do not project them. For area symbols uncovered by the projected area, we project them and connect to other close nodes to them (see the following right figure).



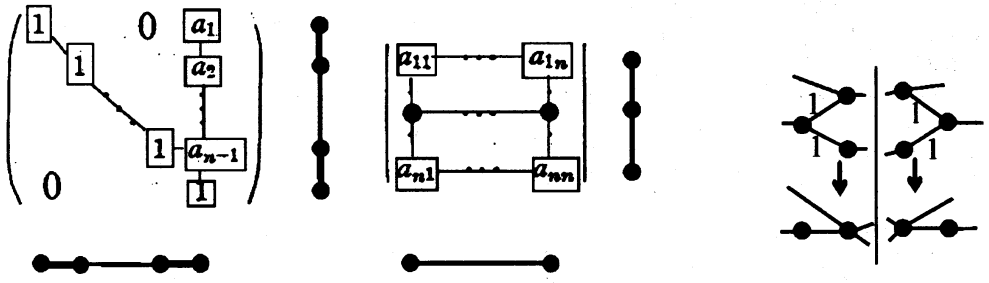
7. Vertical projection of the network

Similarly, we vertically project the network by identifying elements which are vertically connected each other. We also identify elements on the left end (or the right end) of the matrix.

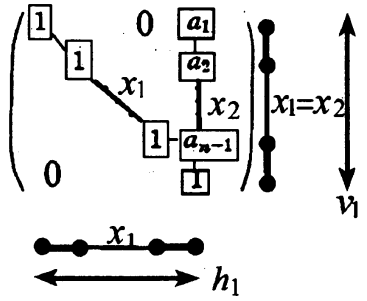


8. Identification of nodes and paths

On each projected network, we identify nodes having a common node at 1-distance from both of them including the directions. Moreover, we identify paths whose origin and terminal are same. We store the information of these identifications.



Thus, we obtain two projected networks. All the lengths of total paths from the upper end (or the left end) to the lower end (resp. the right end) must be equal to the number of rows (resp. columns) of the matrix. We let v be the number of rows and h be the number of columns, and assign an variable to each arc of the path. Then, we can set up the simultaneous system of equations by the lengths from end to end and the information of the paths' identification.



The length of the paths:

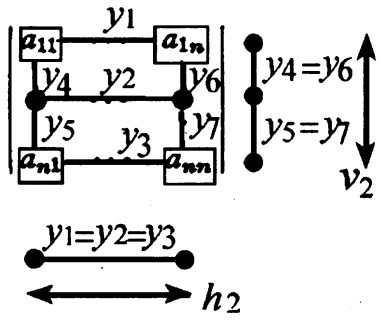
$$\begin{cases} v_1 = x_1 + 2 \\ h_1 = x_1 + 2 \end{cases}$$

The identification information:

$$x_1 = x_2$$

The conditions:

$x_1, x_2 \geq 2,\ v_1, h_1 \geq 0,\ v_1, h_1, x_1, x_2 \in \mathbb{Z}$



The length of the paths:

$$\begin{cases} v_2 = y_4 + y_5 \\ h_2 = y_1 \end{cases}$$

The identification information:

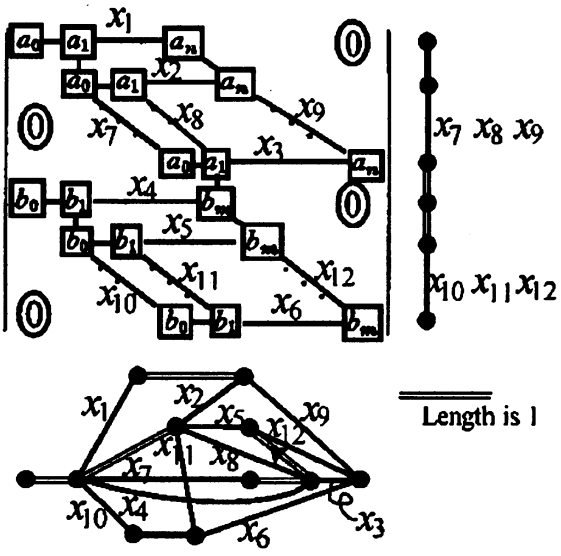
$y_1 = y_2 = y_3,\ y_4 = y_6,\ y_5 = y_7$

The conditions:

$y_1, y_2, y_3 \geq 2, \quad y_4, y_5, y_6, y_7 \geq 1,$
$v_2, h_2 \geq 0, \quad v_2, h_2, y_i \in \mathbb{Z}$

3.4 Structure analysis

By solving the simultaneous system introduced from each projected network so that v and h are minimum, we can obtain the minimum numbers of rows and columns and the relative positions between the connected elements. We gridiron the matrix and put the elements on the grids by obtained values. The area symbols have the connected components separated by paths as their own areas.



The lengths of the paths :

$$v = x_7 + x_{10} + 3 \quad \text{(V-1)}$$

$$\begin{cases} h = x_1 + x_9 + 2 & \text{(H-1)} \\ h = x_2 + x_9 + 2 & \text{(H-2)} \\ h = x_3 + x_8 + 2 & \text{(H-3)} \\ h = x_8 + x_{12} + 3 & \text{(H-4)} \\ h = x_5 + x_{12} + 2 & \text{(H-5)} \\ h = x_3 + x_7 + 2 & \text{(H-6)} \\ h = x_3 + x_4 + 1 & \text{(H-7)} \\ h = x_4 + x_{12} + 2 & \text{(H-8)} \\ h = x_6 + x_{11} + 2 & \text{(H-9)} \\ h = x_6 + x_{10} + 2 & \text{(H-10)} \end{cases}$$

The identification information:

$x_7 = x_8 = x_9, x_{10} = x_{11} = x_{12}.$

The conditions:

$x_i \geq 2 \quad h, v \geq 0,$
$v, h, x_i \in \mathbb{Z}.$

Fig. 6. Example for the structure analysis

We show the algorithm to solve the simultaneous system of equations so that v and h are minimum through Fig. 6

	v	h	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_{10}
(V-1)		×	×	×	×	×	×	×		
(H-1)	×			×	×	×	×	×		×
(H-2)	×		×		×	×	×	×		×
(H-3)	×		×	×		×	×	×		×
(H-4)	×		×	×	×	×	×	×		
(H-5)	×		×	×	×	×		×	×	
(H-6)	×		×	×		×	×	×	×	
(H-7)	×		×	×			×	×	×	×
(H-8)	×		×	×	×		×	×	×	
(H-9)	×		×	×	×	×	×		×	

Table 3. Example of the solution table

In the example, we can delete the equation (H-10) because it becomes (H-9) by using the identification information, $x_{10} = x_{11}$.

In order to solve the system of equations, we use *the solution table* for v and h, whose columns correspond to the equations of v or h, and rows correspond to the variables. For each variable which is not included in a equation, we put a '×' mark in the cell corresponding to the variables and the equation.

1. Evaluate the temporary values by substituting minimum values

We substitute minimum values of the variable. Then, we let the maximum value for v (or h) be a temporary value of v (resp. h), and let the minimum values of the variables included in the equations attaining the maximum value of v or h be their own temporary values. We put each of these temporary values on the equation's row of the solution table, which introduces it.

In the example, the equation (V-1) attains the maximum value of v, 7, and (H-4) attains the maximum value of h, 7. For the variables, x_7 and x_{10}, included in (V-1), and x_8 and x_{12} included in (H-4), we let their temporary values be their own minimum values, 2.

$$v = x_7 + x_{10} + 3 \geq 7 \quad \text{(V-1)}$$
$$h = x_8 + x_{12} + 3 \geq 7 \quad \text{(H-4)}$$

	v	h	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_{10}
(V-1)	7	×	×	×	×	×	×	×	2	2
(H-4)	×	7	×	×	×	×	×	×	2	2

2. Substitution of the temporary values

We substitute the temporary values of v, h and variables obtained in the step 1. We solve the monomial equations changed by this substitution, and let the solutions be the temporary values.

In the example, using the temporary values, $v = h = 7$, $x_7 = x_8 = x_9 = 2$ and $x_{10} = x_{11} = x_{12} = 2$, we solve other equations except for (H-7).

$$\begin{cases} 7 = x_1 + 2 + 2 \Rightarrow x_1 = 3 & \text{(H-1)} \\ 7 = x_2 + 2 + 2 \Rightarrow x_2 = 3 & \text{(H-2)} \\ 7 = x_3 + 2 + 2 \Rightarrow x_3 = 3 & \text{(H-3)} \\ 7 = x_5 + 2 + 2 \Rightarrow x_5 = 3 & \text{(H-5)} \\ 7 = x_3 + 2 + 2 \Rightarrow x_3 = 3 & \text{(H-6)} \\ 7 = x_3 + x_4 + 1 & \text{(H-7)} \\ 7 = x_4 + 2 + 2 \Rightarrow x_4 = 3 & \text{(H-8)} \\ 7 = x_6 + 2 + 2 \Rightarrow x_6 = 3 & \text{(H-9)} \end{cases}$$

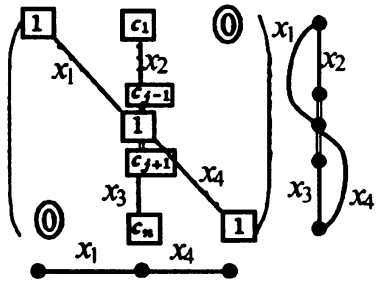
	v	h	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_{10}
(H-1)	×		3	×	×	×	×	×		×
(H-2)	×		×	3	×	×	×	×		×
(H-3)	×		×	×	3	×	×	×		×
(H-4)	×	7	×	×	×	×	×	×	2	2
(H-5)	×		×	×	×	×	3	×	×	
(H-6)	×		×	×	3	×	×	×	×	
(H-7)	×		×	×			×	×	×	×
(H-8)	×		×	×	×	3	×	×	×	
(H-9)	×		×	×	×	×	×	3	×	

Using these temporary values, we repeat this step until new temporary values are not obtained. $7 = h = x_3 + x_4 + 1 = 3 + 3 + 1$ (H-7)

If there are different values of a variable among rows, we let the maximum value among them be the minimum value of the variable, and try from the first step again. The case where both sides of a equation are different constant values never occurs, because the equations are corresponding to paths of certainly existing networks. (In the example, all the values of the variables are determined in this step.)

3. Comparison between two results of v-part and h-part

If there are different values of a variable between v-part and h-part, we let the maximum value between them be the minimum value of the variable, and try from the first step again. For example, the following matrix is this case. In the followings, we use the temporary values on v-part and h-part as common temporary values on them.



$$\begin{cases} v = x_1 + x_4 & \text{(V-1)} \\ v = x_1 + x_3 + 1 & \text{(V-2)} \\ v = x_2 + x_3 + 2 & \text{(V-3)} \\ v = x_2 + x_4 + 1 & \text{(V-4)} \end{cases}$$

$$h = x_1 + x_4 \qquad \text{(H-1)}$$

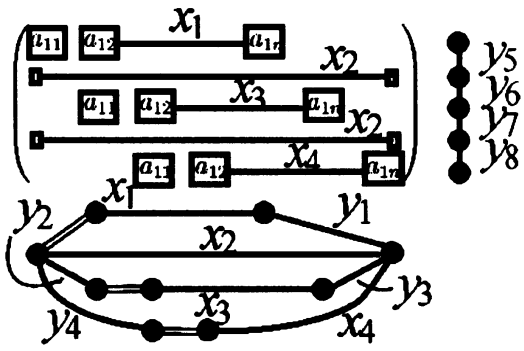
	v	h	x_1	x_2	x_3	x_4
(V-1)		×		×	×	
(V-2)		×	3	×		×
(V-3)	6	×	×	2	2	×
(V-4)		×	×		×	3
(H-1)	×	4	2	×	×	2

4. Substitution of the common temporary values

In a manner similar to the step 2, we substitute the common temporary values for remaining equations, solve monomial equations. If there are different solutions of a variable, we change its minimum value and try from the first step again. We repeat this step until new temporary values are not obtained.

5. Solving remaining equations

After those steps, if some equations are remaining, we obtain the values of the remaining variables by using the elementary transformation of a matrix. The following matrix is this case.



$$v = y_5 + y_6 + y_7 + y_8$$

$$\begin{cases} h = x_1 + y_1 + 1 \\ h = x_2 \\ h = x_3 + y_2 + y_3 + 1 \\ h = x_4 + y_4 + 1 \end{cases} \Rightarrow \begin{cases} v = 4 \\ h = x_2 = 5 \\ x_3 = 2 \\ y_2 = y_3 = y_5 = y_6 = y_7 = y_8 = 1 \end{cases}$$

$$\begin{cases} 5 = x_1 + y_1 + 1 \\ 5 = x_4 + y_4 + 1 \end{cases}$$

$$(x_i \geq 2, y_i \geq 1)$$

From the following simultaneous system (in this case, $m < n$), its *coefficient matrix* is introduced, and then we can deform it into 4-parted matrix by the elementary transformation, where we let r be the rank of the matrix, I_{rr} be a $r \times r$ identify matrix, and O_{pq} be a $p \times q$ zero matrix.

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n = a_1 \\ a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n = a_2 \\ \cdots \\ a_{m1}x_1 + a_{m2}x_2 + \cdots + a_{mn}x_n = a_m \end{cases}$$

$$\Rightarrow \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \cdots & \cdots & \cdots & \cdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{pmatrix} \Rightarrow \left(\begin{array}{c|c} I_{rr} & O_{r,n-r} \\ \hline O_{m-r,r} & O_{m-r,n-r} \end{array}\right)$$

When the values of the variables corresponding to columns whose elements are zero are given, the others are determined (remarking the reshuffle of the columns), because it means that $x_1, \ldots, x_r$ are represented by linear combinations of $x_{r+1}, \ldots, x_n$. Then, we give the minimum values of $x_{r+1}, \ldots, x_n$, and solve the remaining equations in a manner the similar to the step 2.

The following figure represents the result of the structure analysis of Fig. 6 on the grids.

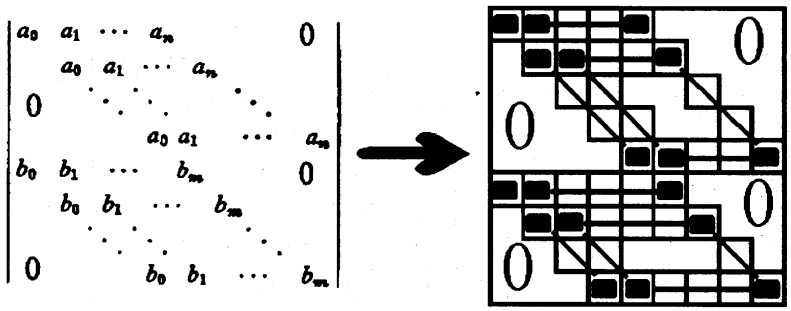


Fig. 7. The result of the structure analysis

4 Experimental results

In order to evaluate our methods, we implemented them into our original OCR System ([3]), named Infty (Fig. 8). Using this system, we evaluated 3 parts of them, the detection of matrices, the segmentation of elements and the construction of networks, because if the 3 parts are exactly completed, the last part, the structure analysis, is also exactly done.

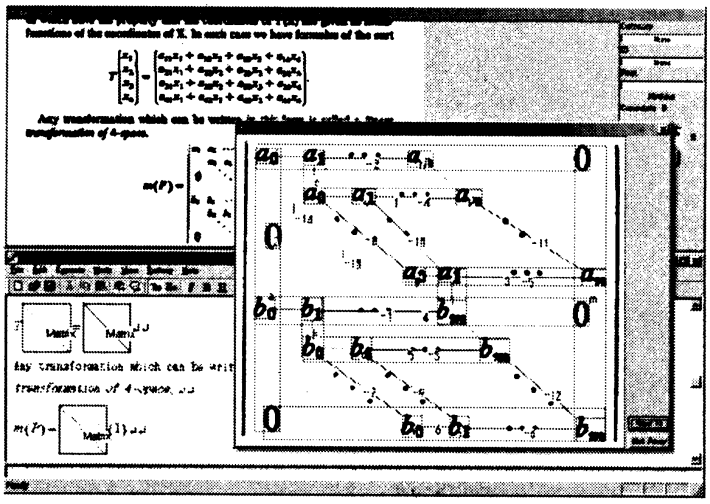


Fig. 8. Image of Infty

We used two English textbooks and two Japanese ones of mathematics including many matrices. For 50 page images of each text (total 200 pages), we counted the numbers of matrices where errors were made with respect to the 3 parts. The 50 page images included about 10 pages where matrices did not appear but big parentheses did in order to evaluate the detection. We show the experimental results below.

Text	M.-1	D.-1	D.-2	E.	M.-2	R.	A.	Conn.	Comp.	P.
E-1	99	1	2	15	83	28	21	4	60	19
E-2	101	1	0	6	94	1	8	1	85	9
E-Total	200	2	2	21	177	29	29	5	145	28
J-1	109	3	1	12	94	7	1	0	86	8
J-2	120	5	3	20	95	1	3	3	92	3
J-Total	229	8	4	32	189	8	4	3	178	11
Total	429	10	6	53	366	37	33	8	323	39

Table 4. Experimental Results 1

Column Names	Meanings
M.-1	The total numbers of matrices
D.-1	The numbers of matrices which could not be found
D.-2	The numbers of detecting formulae or something which are not matrices
E.	The numbers of matrices where the segmentation of elements was not proper
M.-2	The numbers of matrices whose elements were poperly segmented, given by (M.-1) – (D.-1) – (E.)
R.	The numbers of matrices where the extraction of repeat symbols was not proper
A.	The numbers of matrices where the judgment of area symbols was not proper
Conn.	The numbers of matrices where connections between elements were not proper
Comp.	The numbers of matrices whose structures were completely analyzed
P.	The numbers of matrices whose connections were not completely extracted, but positions of elements could be completely analyzed

Table 5. Meanings of columns in Tables 4

Text	D.-R	E.-R	R.-R	A.-R	Conn.-R
E-1	99.0	84.7	66.3	74.7	95.2
E-2	99.0	94.0	98.9	91.5	98.9
E-Total	99.0	89.4	83.6	83.6	97.2
J-1	97.2	88.7	92.6	98.9	100.0
J-2	95.8	82.6	98.9	96.8	96.8
J-Total	96.5	85.5	95.8	97.9	98.4
Total	97.7	87.4	89.9	91.0	97.8

Table 6. Experimental Results 2

Column Names	Meanings
D.-R	The ratio of proper detection of matrices, given by {(M.-1) – (D.-1)}/(M.-1)
E.-R	The ratio of proper segmentation of elements, given by {(M.-1) – (D.-1) – (E.)}/{(M.-1) – (D.-1)}
R.-R	The ratio of proper extraction of repeat symbols, given by {(M.-2) – (R.)}/(M.-2)
A.-R	The ratio of proper judgment of area symbols, given by {(M.-2) – (A.)}/(M.-2)
Conn.-R	The ratio of proper connections between elements, given by {(M.-2) – (Conn.)}/(M.-2)

Table 7. Meanings of columns in Tables 6

Text	Comp.-R1	Comp.-R2	(Comp.+P.)-R1	(Comp.+P.)-R2
E-1	60.6	72.3	79.8	95.2
E-2	84.2	90.4	93.1	100.0
E-Total	72.5	81.9	86.5	97.7
J-1	78.9	91.5	86.2	100.0
J-2	76.7	96.8	79.2	100.0
J-Total	77.7	94.2	82.5	100.0
Total	75.3	88.3	84.4	98.9

Table 8. Experimental Results 3

Column Names	Meanings
Comp.-R1	The ratio of complete analysis of matrix structures to M.-1, given by (Comp.)/(M.-1)
Comp.-R2	The ratio of complete analysis of matrix structures to M.-2, given by (Comp.)/(M.-2)
(Comp.+P.)-R1	The ratio of proper analysis of element positions to M.-1, given by {(Comp.) + (P.)}/(M.-1)
(Comp.+P.)-R2	The ratio of proper analysis of element positions to M.-2, given by {(Comp.) + (P.)}/(M.-2)

Table 9. Meanings of columns in Tables 8

Table 4 shows that the numbers of matrices where errors were made or analysis was succeeded. Table 6 shows that the ratio of success of each part of our method. Table 8 shows that the ratio of the structure analysis. Table 5, 7 and 9 show meanings of the columns of the 3 tables.

Table 6 shows that the detection of matrices has some measure of high accuracy. The misdetection are mainly caused by errors of line segmentation and broken big parentheses by bad conditions of the prints. It was found that the segmentation of elements had a tendency to segment an element including long subscripts into several elements. Table 6 shows that the recognition rates depend on textbooks. One of the main reasons for the dependence is that all textbooks have their own distinctive notations of matrices. Interestingly, error frequency of the judgment of area symbols on simple matrices was higher than on complicated one, because the information of character sizes on simple matrices was less than on complicated one. Comparing (Comp.-R1) with (Comp.-R2) and ((Comp.+P.)-R1) with ((Comp.+P.)-R2) proves that if the segmentation of elements succeeds, the next analysis will be successful at a remarkable rate.

5 Conclusion

We proposed a practical method to recognize matrices containing abbreviation symbols and a format to represent their structure to output the recognition

result. We defined the domain, allowing matrices to contain formula elements, area symbols, and repeat symbols in the matrix coordinate.

The method consists of 4 independent parts; detection of matrices, segmentation of elements, construction of networks and analysis of the matrix structure. In the detection of matrices, we use very simple algorithm using correspondence of big parentheses, and the experimental results prove its high accuracy. To segment characters in a matrix into elements, we use distances between the characters and some characters' features in the mathematical structure. However, the features are not enough to segment, so there are many errors in the experiment. In the construction of networks, we project the connections on a matrix vertically or horizontally. The projected networks are robust against mis-connection between elements on a matrix, namely if there are some lost connections, we can obtain the element positions on a matrix. The construction of networks has some measure of accuracy in the experimental results, but there are some errors in irregular notations. In the analysis of the matrix structure, we let the length of repeat symbols in a matrix be variable. Then, we set up equations for the height and width of the matrix from its vertically projected network and horizontally one. Using the minimum values and ranges of solutions (all solutions are positive integers) instead of the linear programming, we solve the equations and obtain the width and height of a matrix and element positions.

For further improvement, we will try the following problems:

1. To use more mathematical information for improvement of the segmentation of elements.
2. To recognize the decorators of matrices which we excluded in this paper.

References

1. D. Blostein and A. Grbavic, *Recognition of Mathematical Notation*, Handbook of Character Recognition and Document Analysis, Eds. H. Buke, and P. Wang, Word Scientific, 1997.
2. M. Okamoto and H. Twaakyondo, *Structure analysis and recognition of mathematical expressions*, Proceedings of Third International Conference on Document Analysis and Recognition, Wontreal, 1995, pp. 430-437.
3. Y. Eto, M. Sasai and M. Suzuki, *Mathematical formula recognition using virtual link network*, ICDAR 2001.