(12) UK Patent Application (19) GB (11) 2 139 787 A

(43) Application published 14 Nov 1984

(21) Application No 8411256

(22) Date of filing 2 May 1984

(30) Priority data

(31) 58/079341 (32) 9 May 1983 (33) JP

(71) Applicant
Hitachi Ltd. (Japan),
6 Kanda Surugadai 4-chome, Chiyoda-ku, Tokyo, Japan

(72) Inventors
Tsutomu Tashiro,
Koichi Haruna,
Norihisa Komoda

(74) Agent and/or Address for Service
J. A. Kemp & Co.,
14 South Square, Gray's Inn, London WC1R 5EU

(51) INT CL³
G06F 15/46 H03K 13/24

(52) Domestic classification
G4A JR U
U1S 2184 G4A

(56) Documents cited
None

(58) Field of search
G4A
G3N

(54) Facilities control

(57) A condition-responsive control arrangement for facilities, e.g in an automated factory, provides for receiving status signals from the facilities and delivering command signals responsive to the status signals to the facilities. Upon the generation of a status signal, the identity number of a rule set (control logic) for use is determined; a corresponding internal code (Fig. 3*b*) is generated from the status signal on the basis of look-up data stored in memory means; the corresponding coded rules are read out from another memory means on the basis of the identity number of the rule set; the internal code of a final control command is decided by referring to a further memory means (50, Fig. 5B) which stores codes of conclusions formed on the basis of the coded status signals and the applied coded rules; and the command signal for a facility is generated from the internal code of the decided control command by referring to the stored look-up data. The use of internal codes provides faster data processing than in prior art arrangements using character strings.

FIG. 3

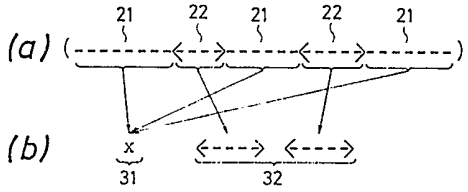
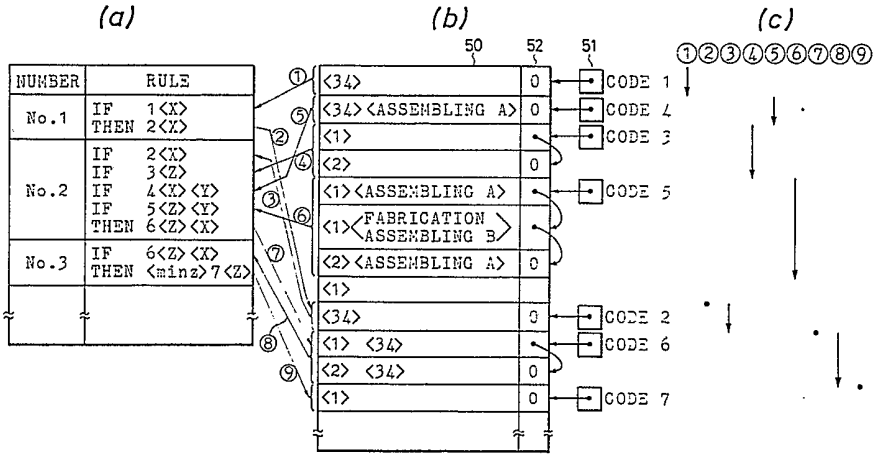


FIG. 5B



GB 2 139 787 A

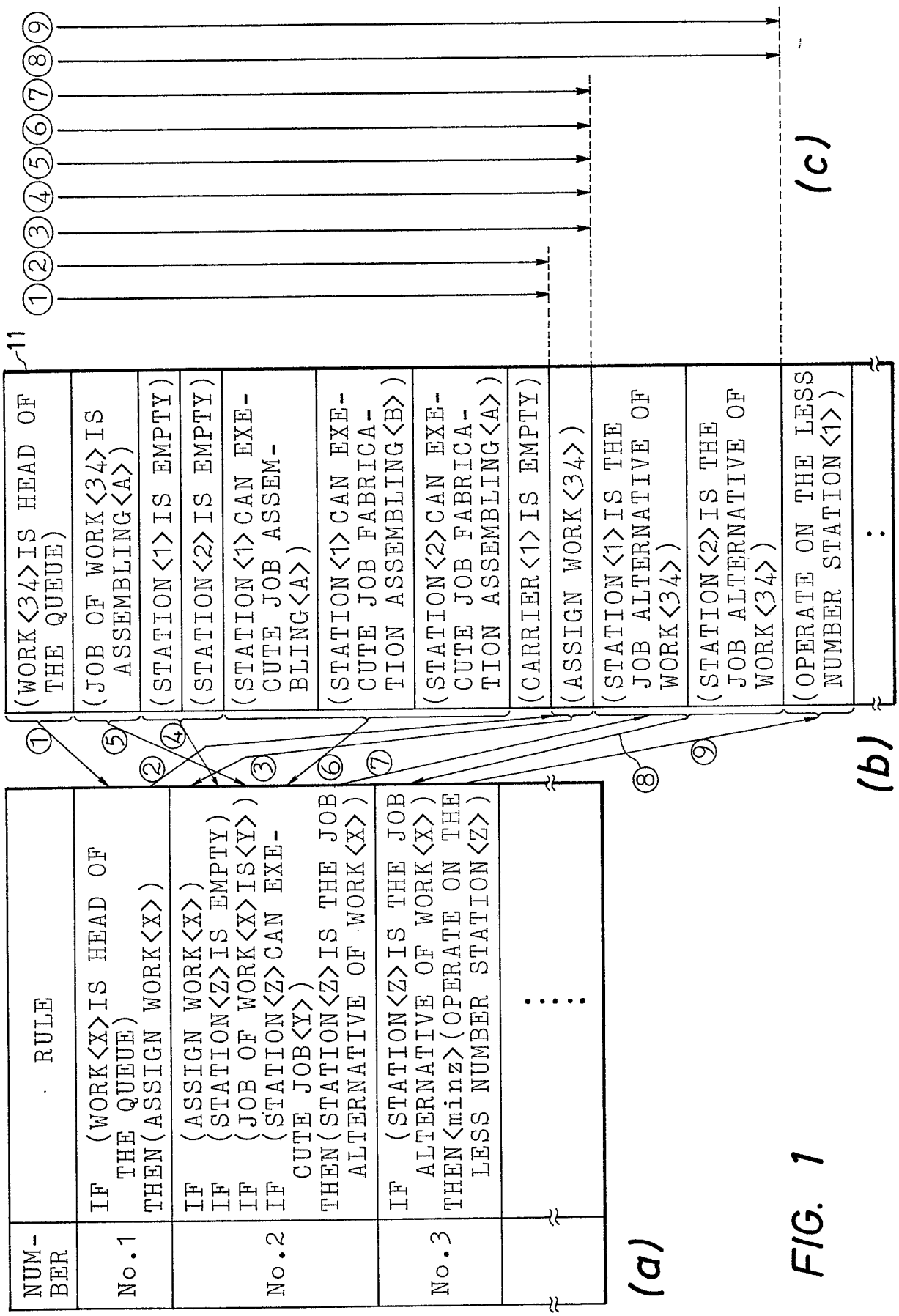
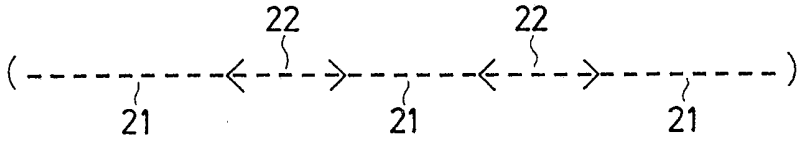

NUM-BER
RULE
No.1
IF (WORK<X>IS HEAD OF THE QUEUE)
THEN(ASSIGN WORK<X>)
No.2
IF (ASSIGN WORK<X>)
IF (STATION<Z>IS EMPTY)
IF (JOB OF WORK<X>IS<Y>)
IF (STATION<Z>CAN EXECUTE JOB<Y>)
THEN(STATION<Z>IS THE JOB ALTERNATIVE OF WORK<X>)
No.3
IF (STATION<Z>IS THE JOB ALTERNATIVE OF WORK<X>)
THEN<minz>(OPERATE ON THE LESS NUMBER STATION<Z>)
(a)
11
(WORK<34>IS HEAD OF THE QUEUE)
(JOB OF WORK<34>IS ASSEMBLING<A>)
(STATION<1>IS EMPTY)
(STATION<2>IS EMPTY)
(STATION<1>CAN EXECUTE JOB ASSEMBLING<A>)
(STATION<1>CAN EXECUTE JOB FABRICATION ASSEMBLING<B>)
(STATION<2>CAN EXECUTE JOB FABRICATION ASSEMBLING<A>)
(CARRIER<1>IS EMPTY)
(ASSIGN WORK<34>)
(STATION<1>IS THE JOB ALTERNATIVE OF WORK<34>)
(STATION<2>IS THE JOB ALTERNATIVE OF WORK<34>)
(OPERATE ON THE LESS NUMBER STATION<1>)
(b)
(c)

FIG. 1

FIG. 2

(a)



(b)

FORM	IF PART	THEN PART
1	IF (·····<CONSTANT>·····<CONSTANT>·····)	THEN (·····<CONSTANT>·····)
2	IF (·····<VARIABLE>·····<CONSTANT>·····)	THEN (·····<VARIABLE>·····<VARIABLE>·····)
3	IF (·····<VARIABLE>·····<VARIABLE>·····)	THEN <PROCEDURE NAME> (····<VARIABLE>····)

2139787

2139787

FIG. 3

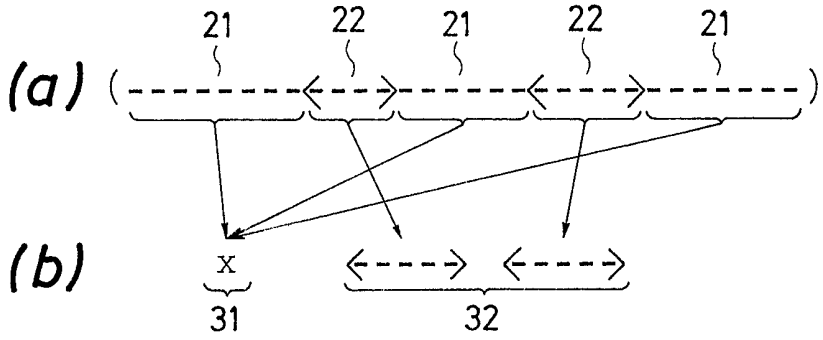


FIG. 4

NUMBER	RULE
No.1	IF 1 <X> THEN 2 <X>
No.2	IF 2 <X> IF 3 <Z> IF 4 <X> <Y> IF 5 <Z> <Y> THEN 6 <Z> <X>
No.3	IF 6 <Z> <X> THEN <minz> 7 <Z>
	≀

FIG. 5A

(a)

x <1>	x <7>
x <2>	y <13> <1>
x <5>	y <10> <3>
x <6>	y <9> <4>

(b)

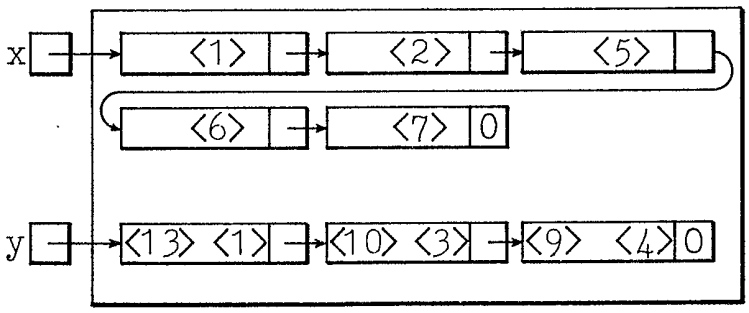


FIG. 5B

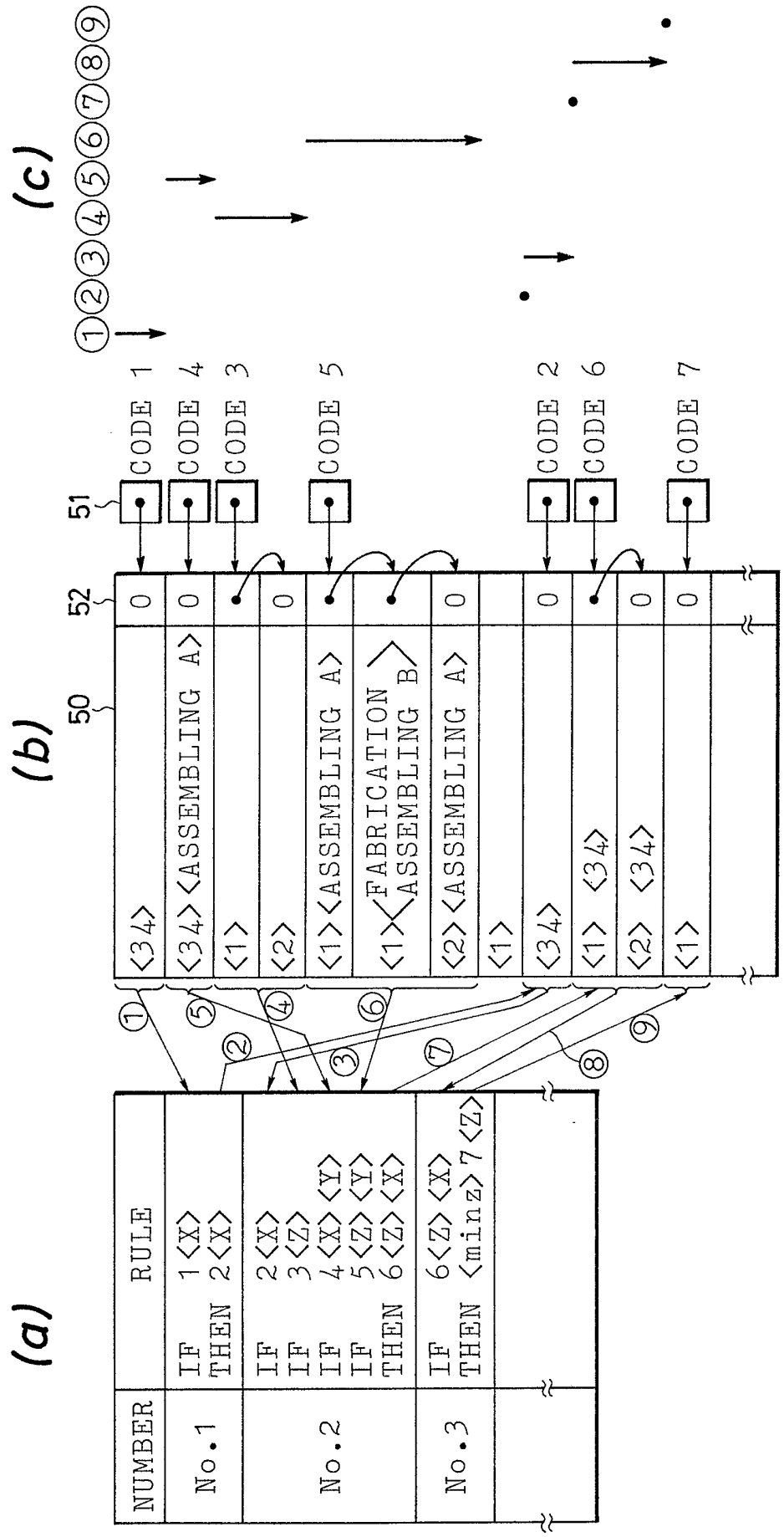

(a)
NUMBER
RULE
No.1
IF 1 <X>
THEN 2 <X>
No.2
IF 2 <X>
IF 3 <Z>
IF 4 <X> <Y>
IF 5 <Z> <Y>
THEN 6 <Z> <X>
No.3
IF 6 <Z> <X>
THEN <minz> 7 <Z>
(b)
50
52
51
<3 4>
<3 4> <ASSEMBLING A>
<1>
<2>
<1> <ASSEMBLING A>
<1> FABRICATION / ASSEMBLING B
<2> <ASSEMBLING A>
<1>
<3 4>
<1> <3 4>
<2> <3 4>
<1>
0
CODE 1
CODE 4
CODE 3
CODE 5
CODE 2
CODE 6
CODE 7
(c)
①②③④⑤⑥⑦⑧⑨

FIG. 6

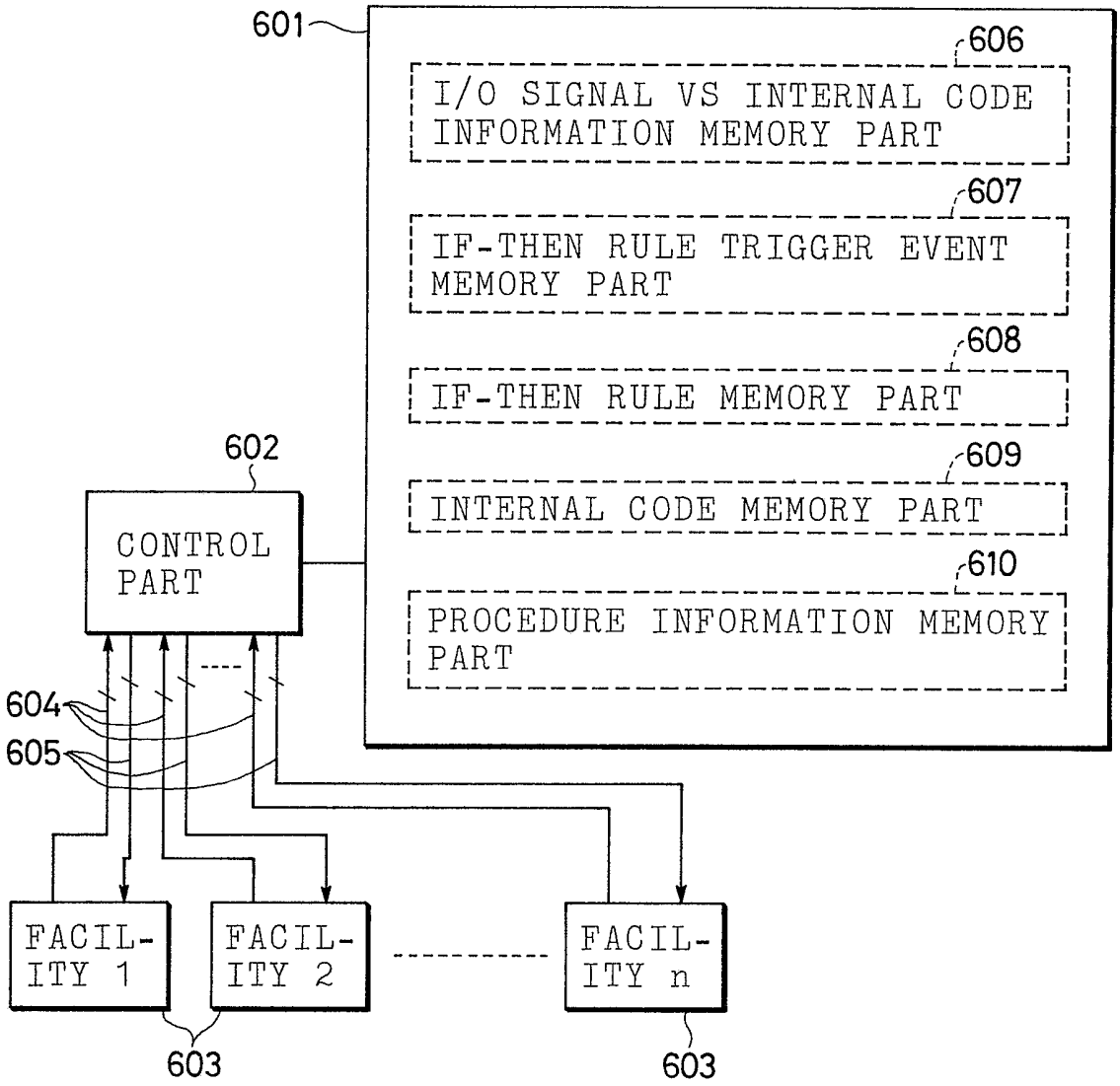

601
606
I/O SIGNAL VS INTERNAL CODE INFORMATION MEMORY PART
607
IF-THEN RULE TRIGGER EVENT MEMORY PART
608
IF-THEN RULE MEMORY PART
609
INTERNAL CODE MEMORY PART
610
PROCEDURE INFORMATION MEMORY PART
602
CONTROL PART
604
605
FACIL-ITY 1
FACIL-ITY 2
FACIL-ITY n
603
603

FIG. 7

(a)

71

ENTRY OF STATUS SIGNAL LINE	NUMERIC CODE	PARAMETER PART
ADDRESS A	1	X
ADDRESS B	8	1

(b)

72

ENTRY TO COMMAND SIGNAL LINE	NUMERIC CODE	PARAMETER PART
ADDRESS C	9	1 X Z
ADDRESS D	10	<1><ASSEM-BLING A>

FIG. 8

(a)

81

ENTRY OF STATUS SIGNAL LINE	NUMBER OF RULES TO BE TRIGGERED
ADDRESS A	1
ADDRESS E	2

(b)

82

1

FIG. 9

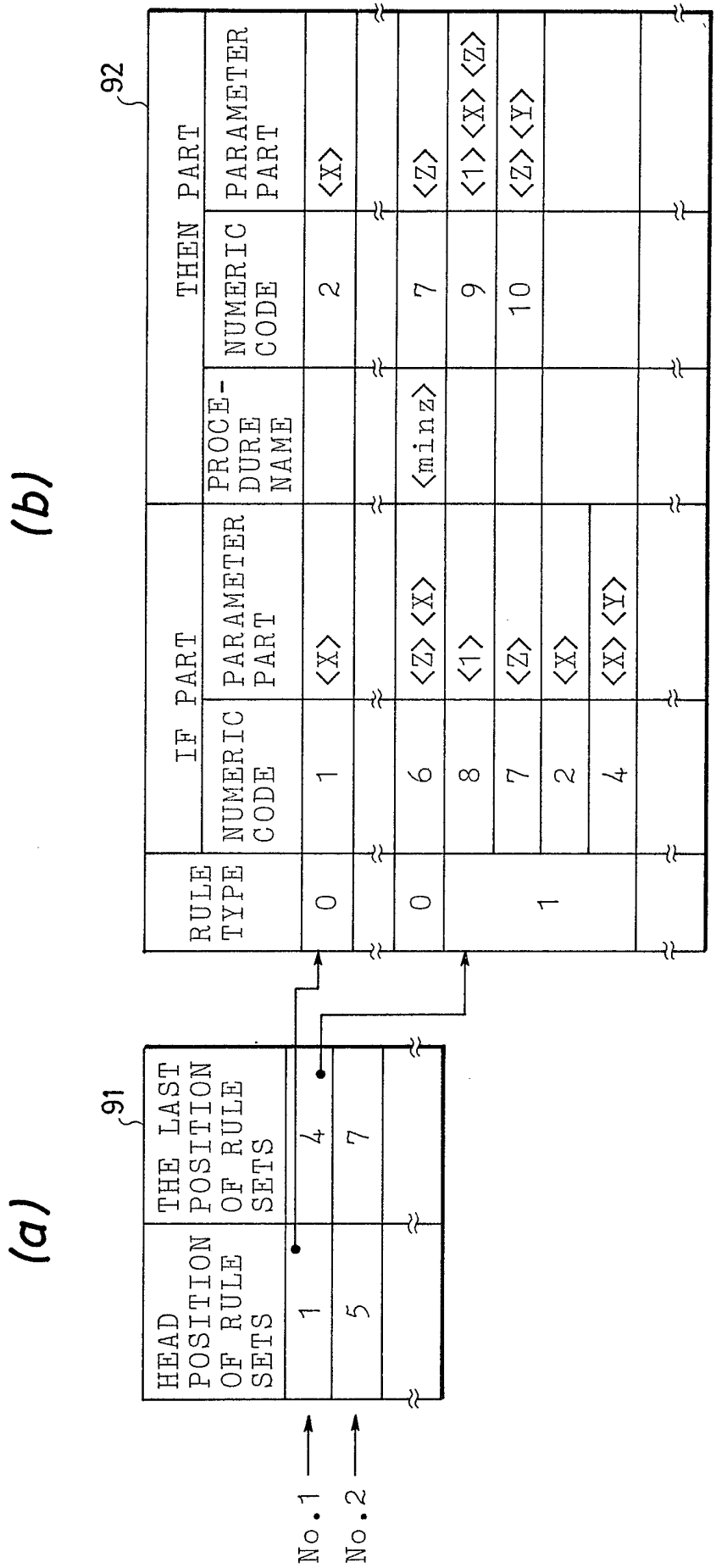
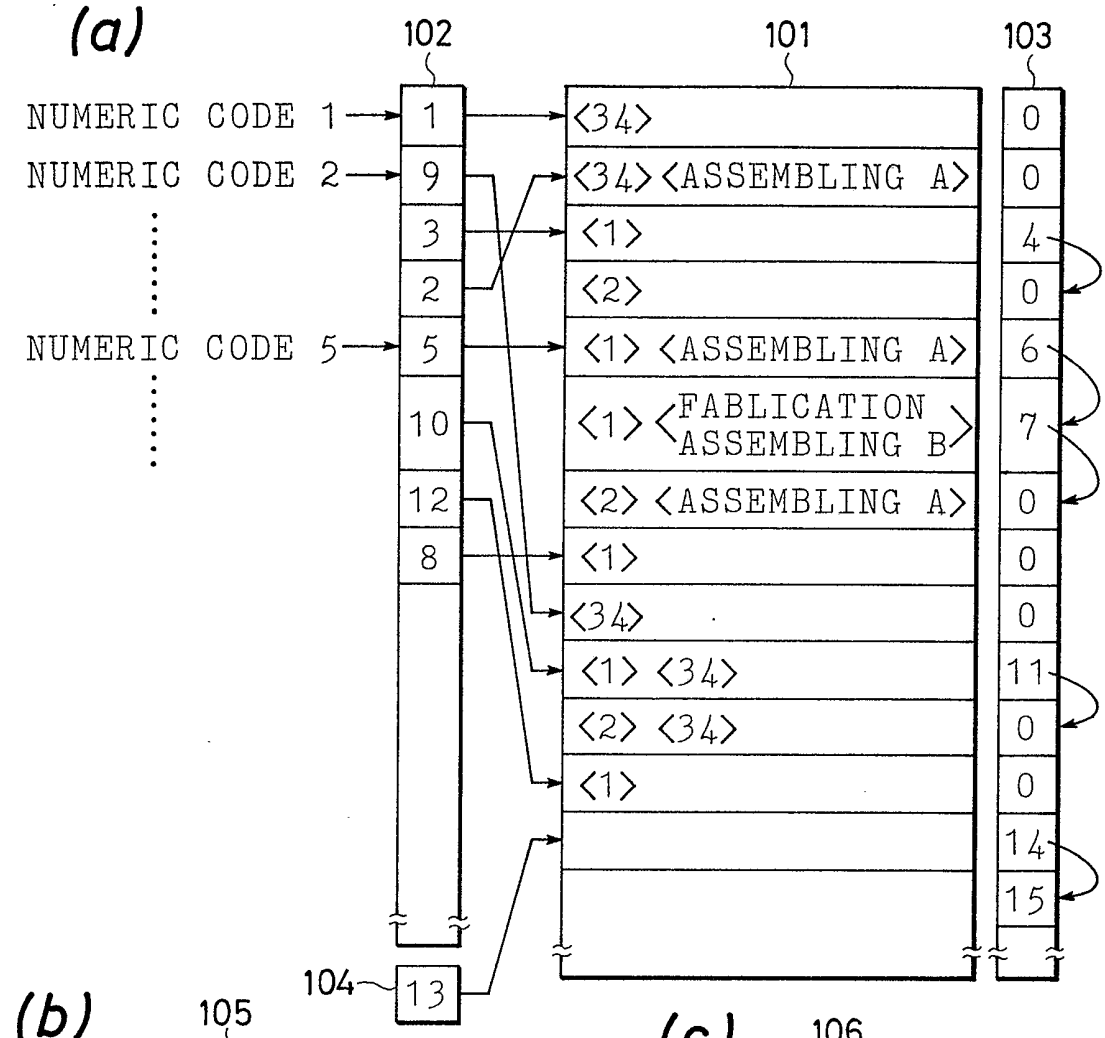

(a)
(b)
91
92
HEAD POSITION OF RULE SETS
THE LAST POSITION OF RULE SETS
1
4
5
7
No.1
No.2
RULE TYPE
IF PART
NUMERIC CODE
PARAMETER PART
PROCE-DURE NAME
THEN PART
NUMERIC CODE
PARAMETER PART
0
1
<X>
2
<X>
0
6
<Z> <X>
<minz>
7
<Z>
1
8
<1>
9
<1> <X> <Z>
7
<Z>
10
<Z> <Y>
2
<X>
4
<X> <Y>

FIG. 10



(b)

105

NUMERIC CODE	PARAMETER PART
9	<1> <34> <1>
10	<1> <ASSEM-BLING A>

(c)

106

W	X	Y	Z
			2

(d)

107

W	X	Y	Z
		ASSEMBLING A	1
		FABRICATION ASSEMBLING B	1
		ASSEMBLING A	2

FIG. 11

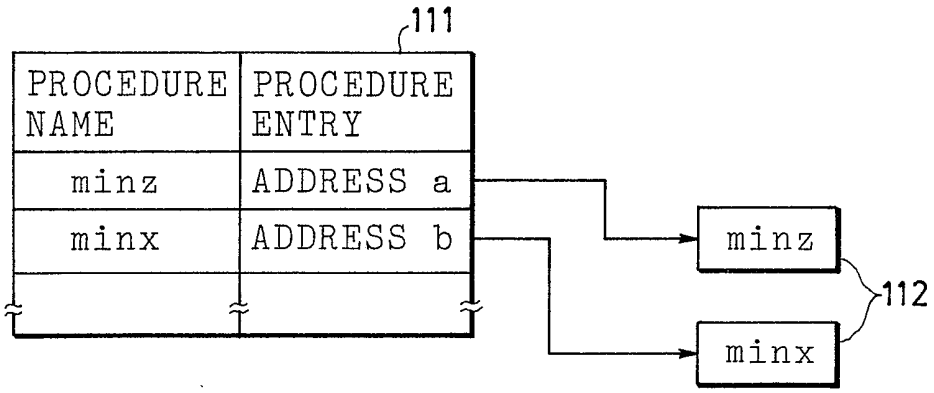


FIG. 12

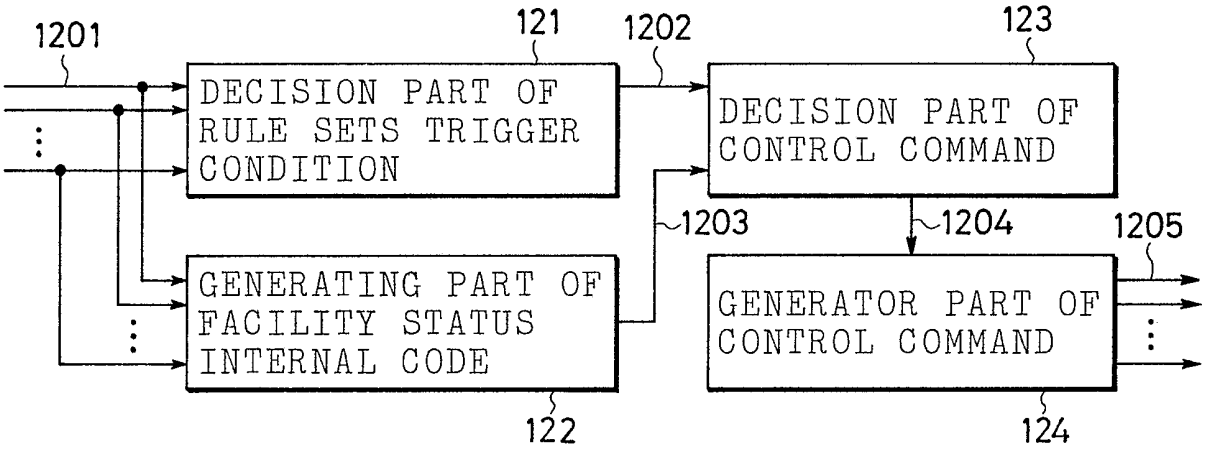


FIG. 17

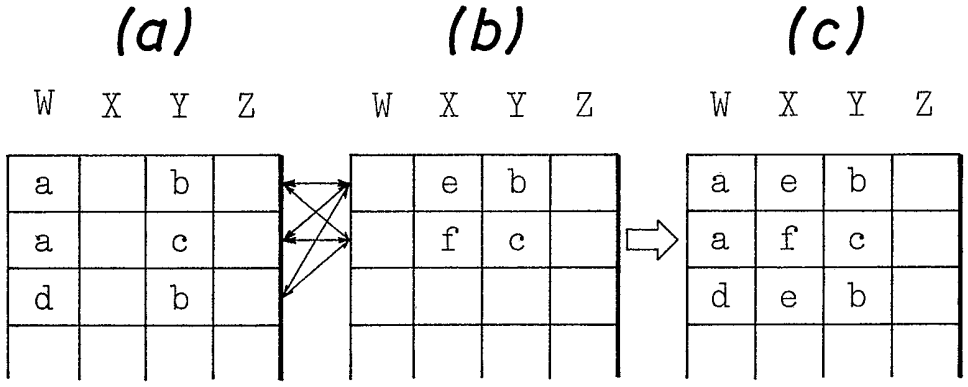


FIG. 13

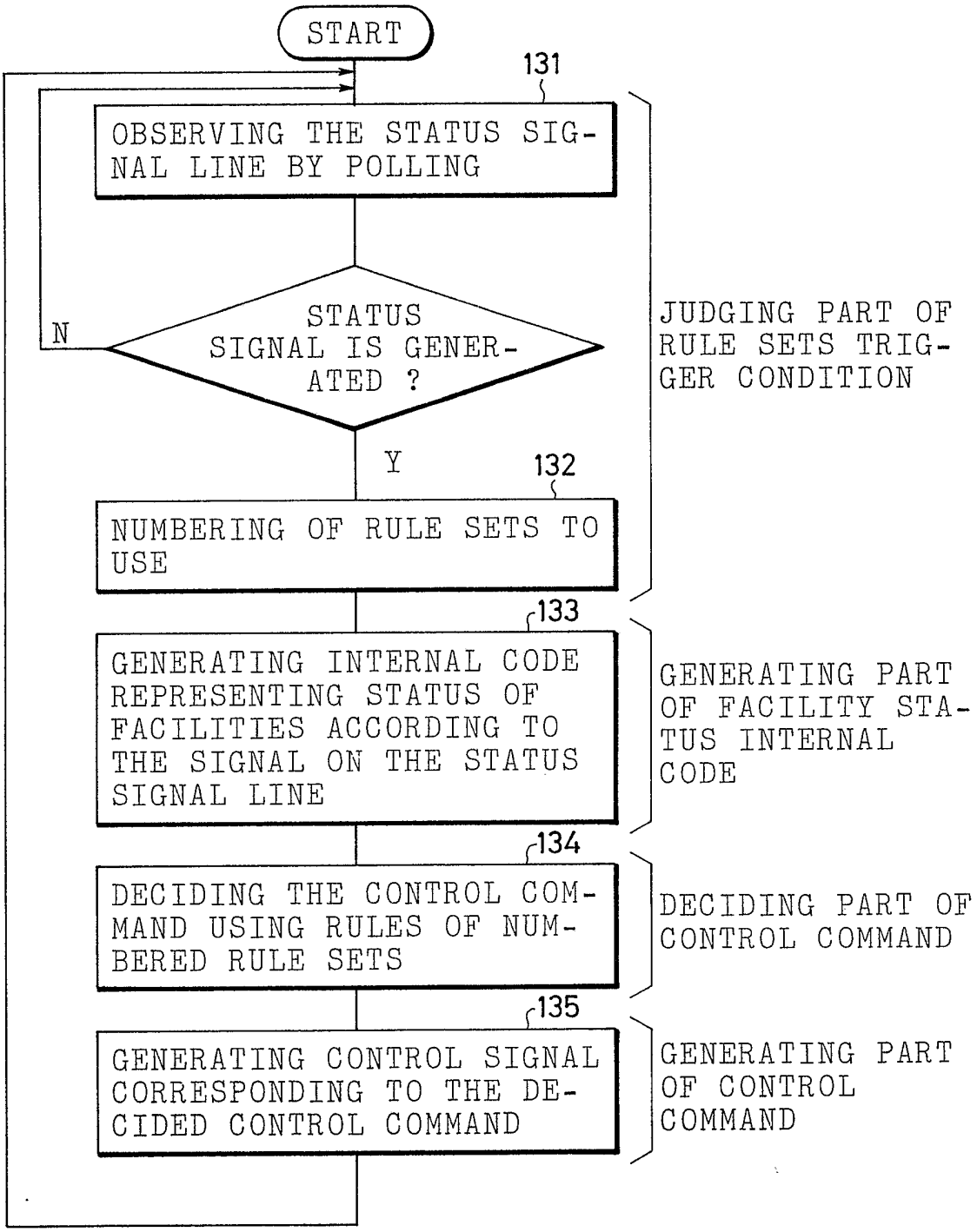

START
131
OBSERVING THE STATUS SIGNAL LINE BY POLLING
STATUS SIGNAL IS GENERATED ?
N
Y
132
NUMBERING OF RULE SETS TO USE
133
GENERATING INTERNAL CODE REPRESENTING STATUS OF FACILITIES ACCORDING TO THE SIGNAL ON THE STATUS SIGNAL LINE
134
DECIDING THE CONTROL COMMAND USING RULES OF NUMBERED RULE SETS
135
GENERATING CONTROL SIGNAL CORRESPONDING TO THE DECIDED CONTROL COMMAND
JUDGING PART OF RULE SETS TRIGGER CONDITION
GENERATING PART OF FACILITY STATUS INTERNAL CODE
DECIDING PART OF CONTROL COMMAND
GENERATING PART OF CONTROL COMMAND

FIG. 14

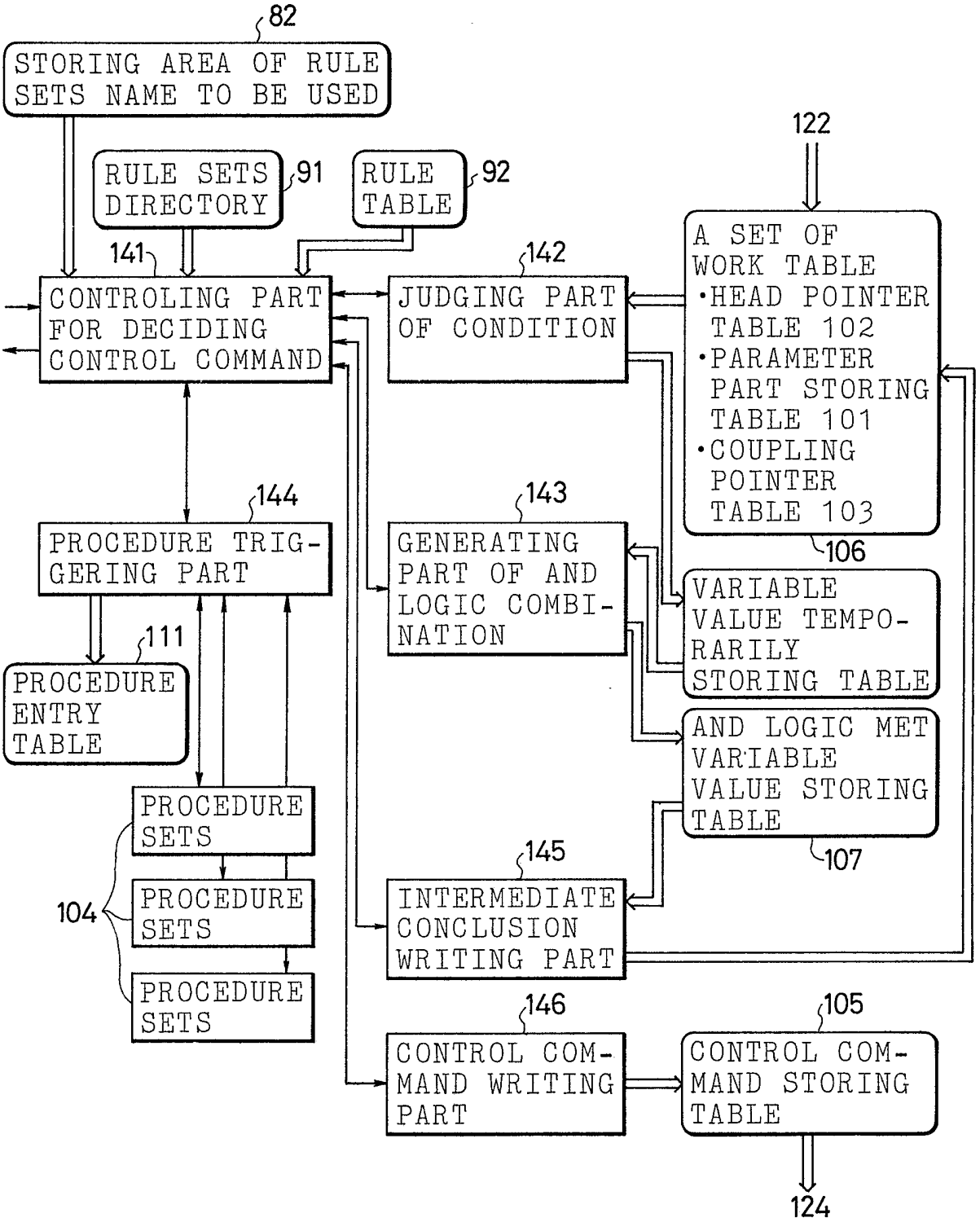

82
STORING AREA OF RULE SETS NAME TO BE USED
RULE SETS DIRECTORY
91
RULE TABLE
92
122
141
CONTROLING PART FOR DECIDING CONTROL COMMAND
142
JUDGING PART OF CONDITION
A SET OF WORK TABLE
·HEAD POINTER TABLE 102
·PARAMETER PART STORING TABLE 101
·COUPLING POINTER TABLE 103
106
144
PROCEDURE TRIGGERING PART
143
GENERATING PART OF AND LOGIC COMBINATION
VARIABLE VALUE TEMPORARILY STORING TABLE
111
PROCEDURE ENTRY TABLE
AND LOGIC MET VARIABLE VALUE STORING TABLE
107
PROCEDURE SETS
PROCEDURE SETS
PROCEDURE SETS
104
145
INTERMEDIATE CONCLUSION WRITING PART
146
CONTROL COMMAND WRITING PART
105
CONTROL COMMAND STORING TABLE
124

FIG. 15A

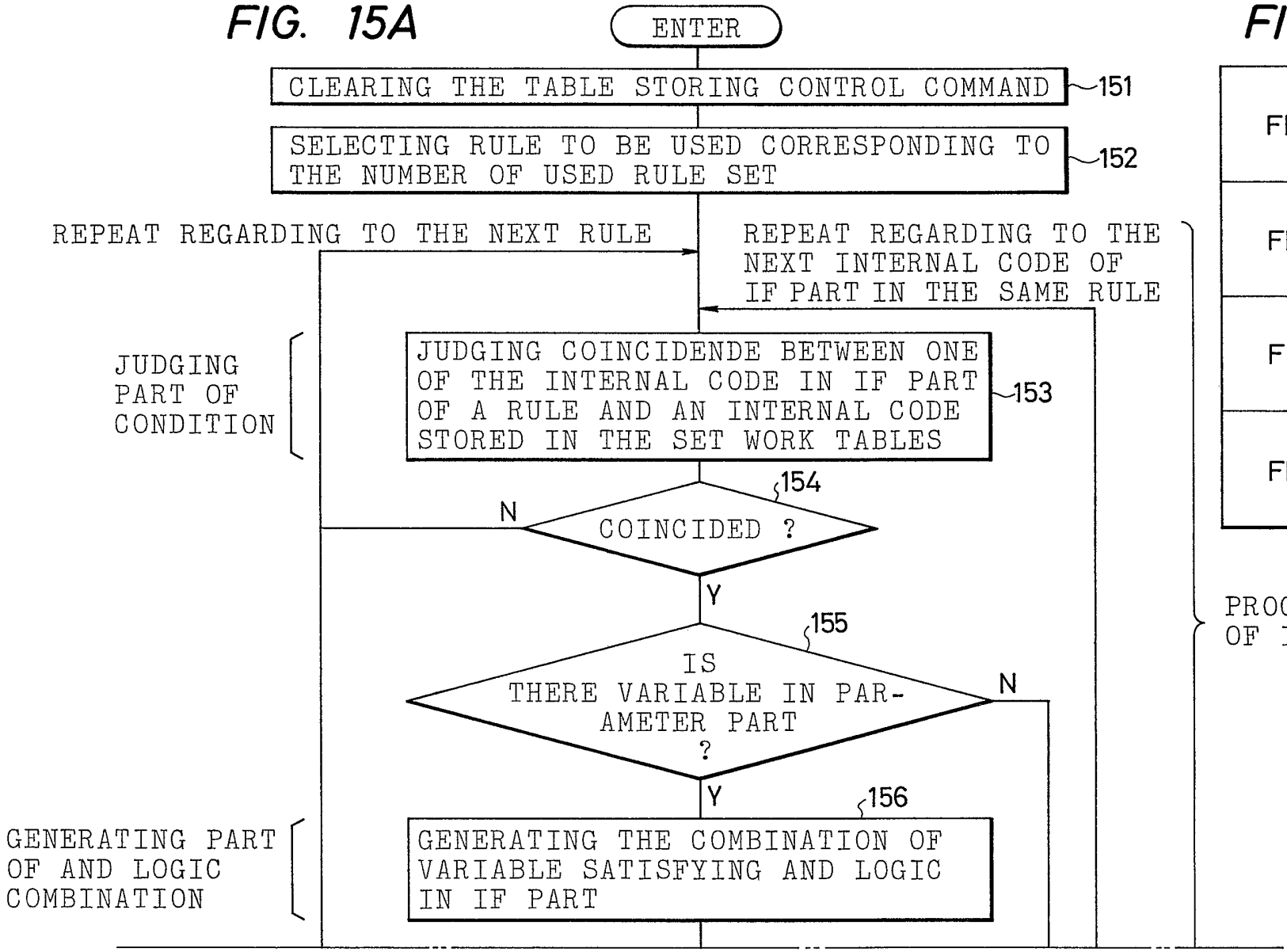
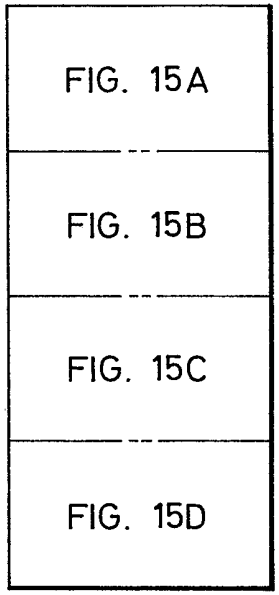


FIG. 15



2139787

FIG. 15B

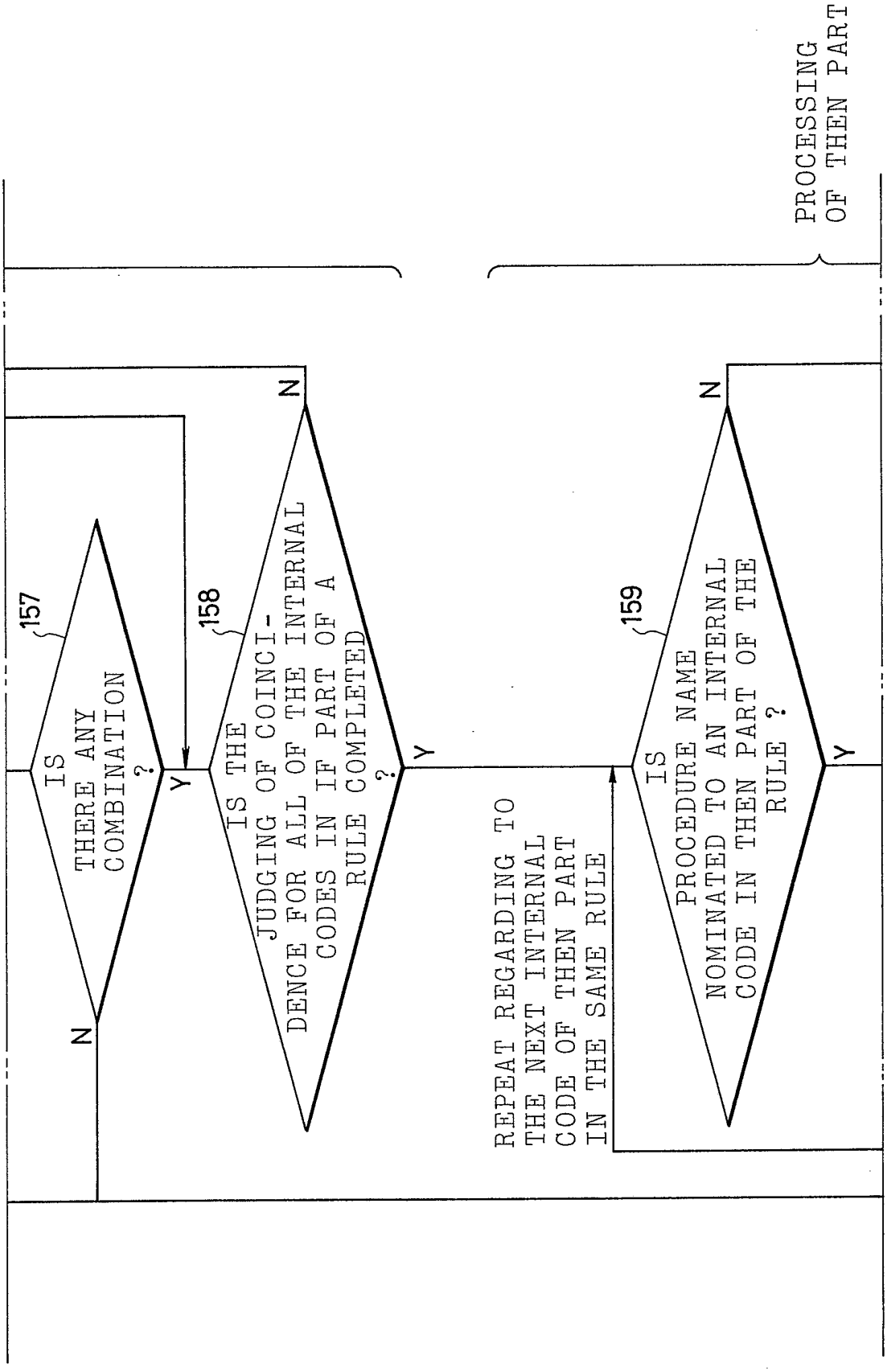

PROCESSING OF THEN PART
157
IS THERE ANY COMBINATION ?
Y
N
158
IS THE JUDGING OF COINCIDENCE FOR ALL OF THE INTERNAL CODES IN IF PART OF A RULE COMPLETED ?
Y
N
159
IS PROCEDURE NAME NOMINATED TO AN INTERNAL CODE IN THEN PART OF THE RULE ?
Y
N
REPEAT REGARDING TO THE NEXT INTERNAL CODE OF THEN PART IN THE SAME RULE

FIG. 15C

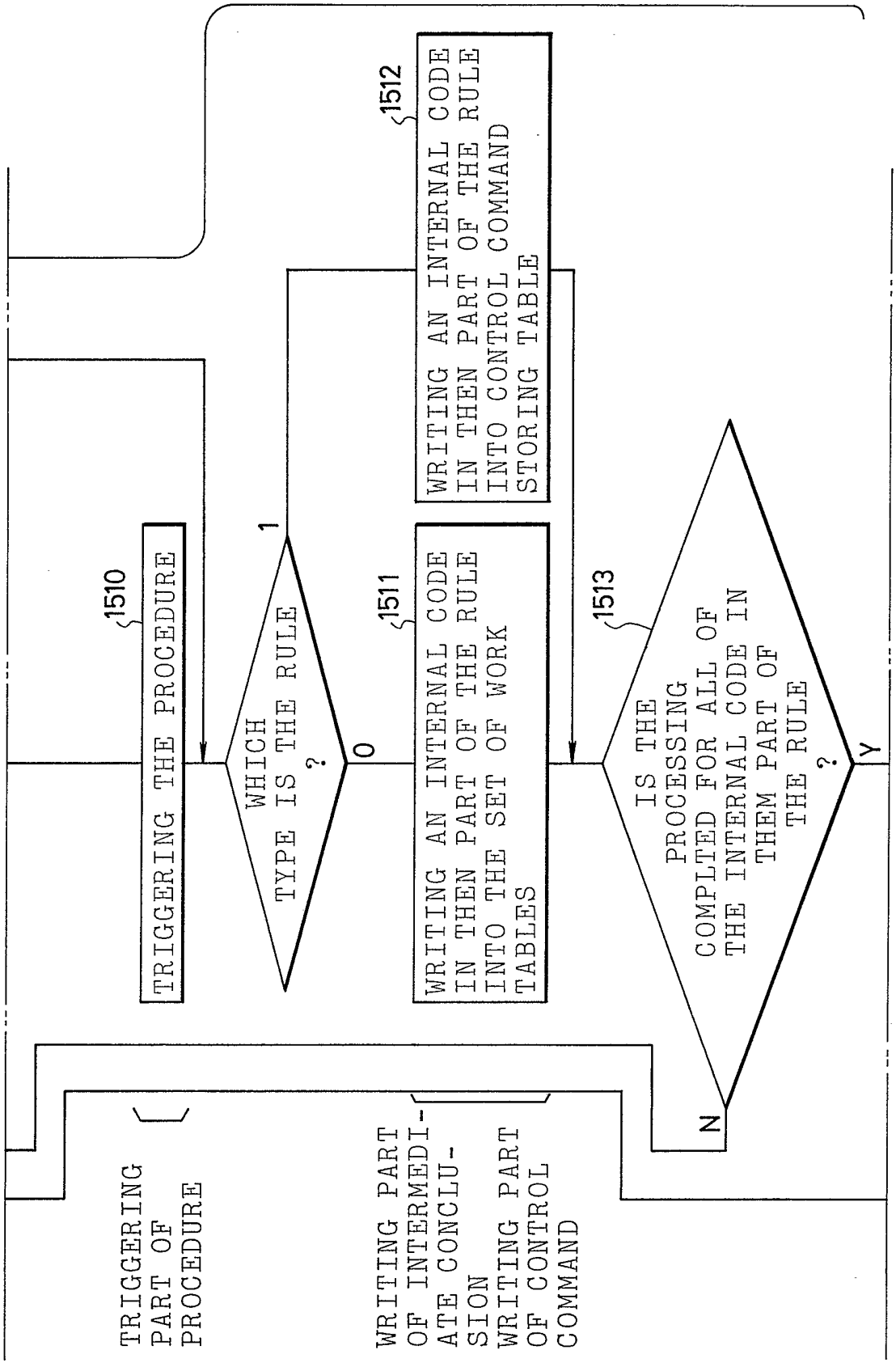

TRIGGERING PART OF PROCEDURE
WRITING PART OF INTERMEDIATE CONCLUSION WRITING PART OF CONTROL COMMAND
1510
TRIGGERING THE PROCEDURE
WHICH TYPE IS THE RULE ?
1
0
1511
WRITING AN INTERNAL CODE IN THEN PART OF THE RULE INTO THE SET OF WORK TABLES
1512
WRITING AN INTERNAL CODE IN THEN PART OF THE RULE INTO CONTROL COMMAND STORING TABLE
1513
IS THE PROCESSING COMPLTED FOR ALL OF THE INTERNAL CODE IN THEM PART OF THE RULE ?
Y
N

FIG. 15D

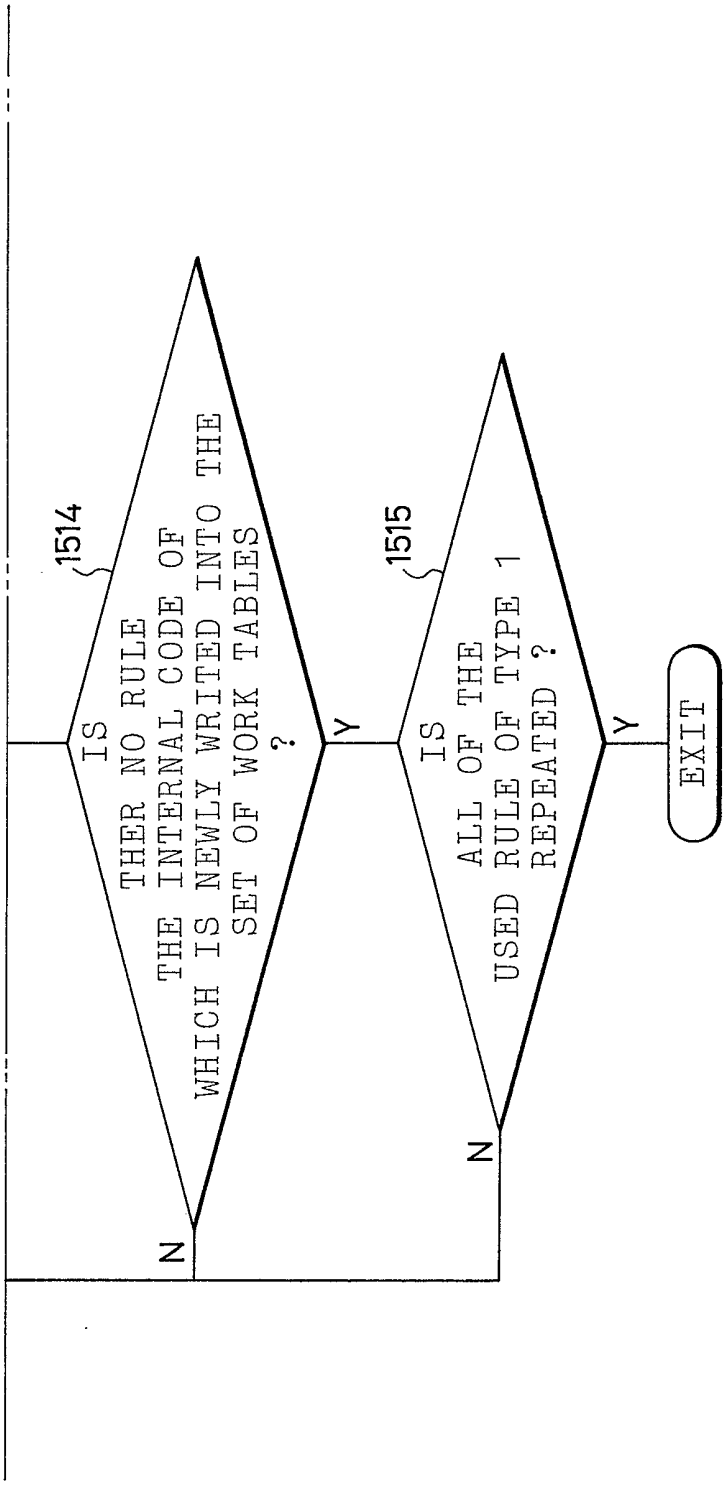
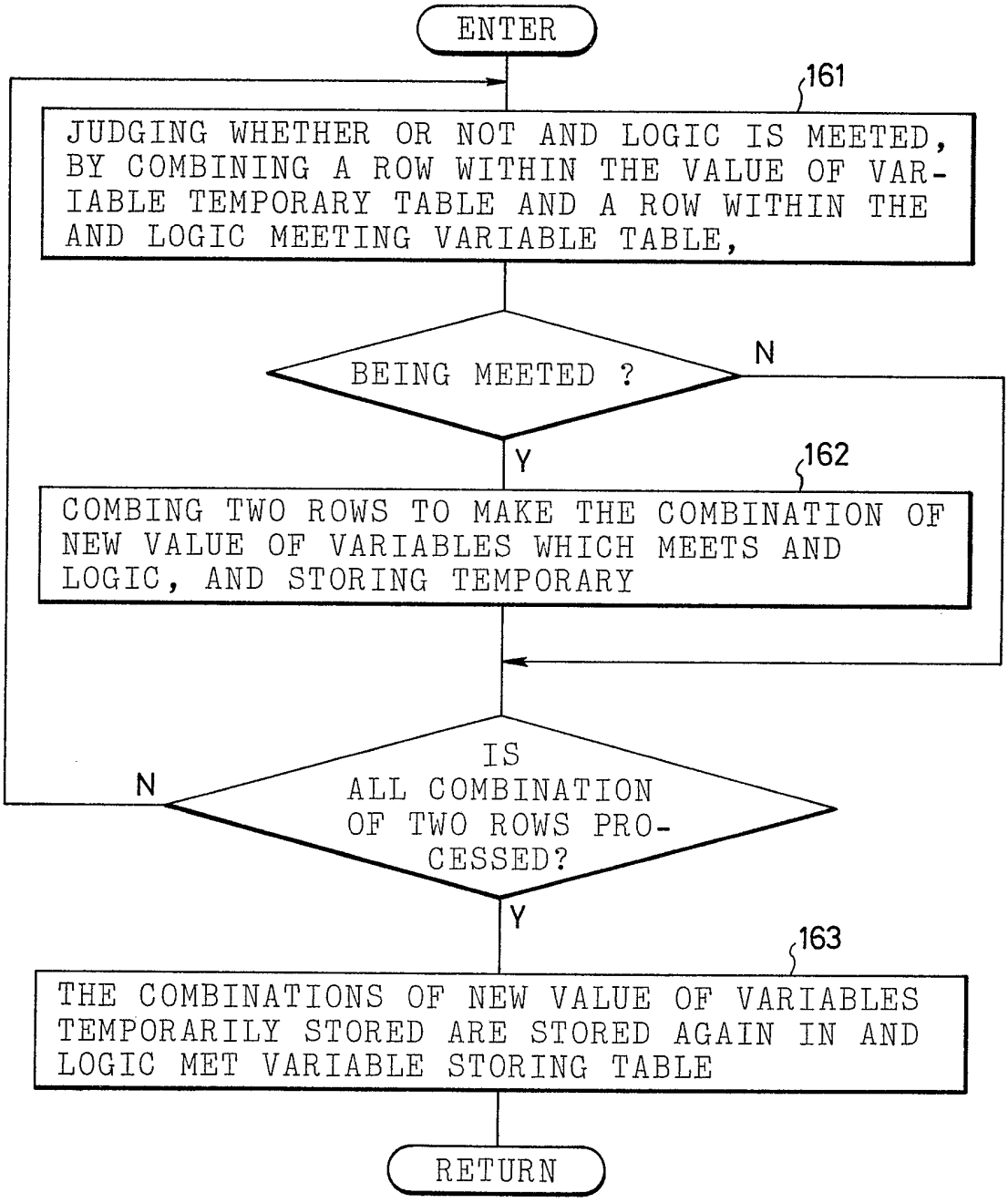

1514
IS THER NO RULE THE INTERNAL CODE OF WHICH IS NEWLY WRITED INTO THE SET OF WORK TABLES ?
Y
N
1515
IS ALL OF THE USED RULE OF TYPE 1 REPEATED ?
Y
N
EXIT

FIG. 16


ENTER
161
JUDGING WHETHER OR NOT AND LOGIC IS MEETED, BY COMBINING A ROW WITHIN THE VALUE OF VARIABLE TEMPORARY TABLE AND A ROW WITHIN THE AND LOGIC MEETING VARIABLE TABLE,
BEING MEETED ?
N
Y
162
COMBING TWO ROWS TO MAKE THE COMBINATION OF NEW VALUE OF VARIABLES WHICH MEETS AND LOGIC, AND STORING TEMPORARY
IS ALL COMBINATION OF TWO ROWS PROCESSED?
N
Y
163
THE COMBINATIONS OF NEW VALUE OF VARIABLES TEMPORARILY STORED ARE STORED AGAIN IN AND LOGIC MET VARIABLE STORING TABLE
RETURN

SPECIFICATION

Facilities control method

Background of the invention

The present invention relates to a facilities control method. More particularly, it relates to a facilities control method performing a condition discriminative control in which the statuses of a plurality of facilities in an automated factory or the like are put in a computer, conditions are judged in accordance with predetermined control logics, and when the conditions have held, control commands specified by the control logics are issued to the respective facilities.

Description of the Prior Art:

In an FA (Factory Automation) system, it is desirable that the alterations of manufacturing process facilities attendant upon the change of the quantity of articles to be produced or the change of the product kind itself can be done simply and rapidly.

The Applicant has therefore proposed "Facilities Control Method" easy of the performance and alterations of a facilities control, which has been applied for patents in U.S. etc. Since this method processes character strings by storing them in a work table as they are, the processing is simple. However, the processing time is too long, and the method has had the disadvantage of being unusable for a controlled system which requires a quick control response.

In the above method, control logics are described as a plurality of condition – conclusion parts called IF – THEN rules, each of which consists of a condition part (IF part) descriptive of a condition to be judged and a conclusion part (THEN part) descriptive of a control content to be executed or a content to be presumed when the condition is met. This method permits the easy performance of the condition discriminative control for such reasons (i) that the IF – THEN rules can be stated with any desired character strings, or in Japanese, and (ii) that merely by itemizing the condition – conclusion pairs by the IF – THEN rules, the control content to be executed can be decided automatically by applying the conclusion of any rule to the condition judgemen of another rule.

In order to realize the items (i) and (ii), the facilities control method adopts a method in which the control content to be finally executed is decided by repeating the processing of first converting the statuses of the facilities into the character string which is stored in the work table, subsequently judging the condition of the IF part of the rule by the pattern matching between the character string of the IF part and the character string in the work table, and writing the character string of the THEN part as the conclusion into the work table anew when the condition is met.

This method is simple in the processing, but it involves problems as stated below:

(i) Since the pattern matching is performed using the whole character string as it is, the processing requires a long time.

(ii) In judging the coincidence of the character string of the IF part and writing the character string of the THEN part, all the character strings stored in a character string storing table are checked, so that a large number of useless pattern matching operations are performed.

For these reasons, in the condition discriminative control, a considerable processing time is needed since the reception of the statuses of the facilities till the generation of control commands. It has accordingly been impossible to apply the above method to the controlled system of which a quick control response is required.

Summary of the invention

An object of the present invention is to solve such problems, and to provide a facilities control method which can easily perform a condition discriminative control and which is applicable even to the control of a controlled system requiring a quick control response.

In order to accomplish the object, the present invention provides a facilities control method having a rule memory part which stores rules consisting of conditions and conclusions of facilities, a status memory part which stores statuses of the facilities, job contents to be executed and contents concluded by the rules, and a rule applying part which compares the information stored in the status memory part and the conditions stored in the rule memory part and which stores the conclusions of the rules having had the conditions met into the status memory part, thereby to decide control commands for the facilities; characterized in that information items which represent the statuses of the facilities, the job contents to be executed by the facilities and the contents concluded by the rules are expressed as code information, that they are coupled by pointers for the same codes and thus stored in said status memory part, and that said rule applying part performs comparison processing by tracing said pointers.

Brief description of the drawings

Figure 1 is an explanatory diagram of a control command deciding system proposed previously;

Figure 2 is an explanatory diagram of the forms of IF – THEN rules;

Figure 3 is an explanatory diagram of a method of turning a character string into an internal code, illustrative of the principle of the present invention;

Figure 4 is a diagram showing examples of IF – THEN rules turned into internal codes by the method of Figure 3;

Figures 5A and *5B* are diagrams for explaining a case where internal codes in Figure 1 are stored into a work table by the method of the present invention;

Figure 6 is a block arrangement diagram of a facilities control apparatus showing an embodiment of the present invention;

Figure 7 is a detailed diagram of an input/output signal versus internal code information memory part;

Figure 8 is a detailed diagram of an IF – THEN rule trigger event memory part;

Figure 9 is a detailed diagram of an IF – THEN rule memory part;

Figure 10 is a detailed diagram of an internal code memory part;

Figure 11 is a detailed diagram of a procedure information memory part;

Figure 12 is a functional block diagram of a control part;

Figure 13 is an operating flow chart of the control part;

Figure 14 is a diagram showing the relationships among the functional blocks of a control command deciding part and various tables;

Figure 15 is an operating flow chart of the control command deciding part;

Figure 16 is an operating flow chart of an AND logic combination generating part; and

Figure 17 is a diagram showing examples of creation of the combinations of variables meeting AND logic which are created in the AND logic combination generating part.

Detailed description of the preferred embodiments

For the sake of comparisons with the present invention, the control command deciding method based on the foregoing facilities control method will be first described with reference to Figure 1.

In Figure 1, (a) indicates rules which are some of rules for assigning job works to the work stations of an automated factory, (b) a character string storing table, and (c) the search ranges of the table.

IF – THEN rules include three sorts of forms, the processing methods of which are respectively different. This will be described later.

First, the statuses of facilities are written into the character string storing table 11 (from (work < 34 > is the head of a queue) to (carrier < 1 > is empty)). Next, the pattern matching between (work < X > is the head of a queue) which is the character string of the IF part of rule No. 1 and a character string in the character string storing table 11 is performed (arrow ①). The matching is performed as to places other than those parts of the character strings which are enclosed with < >. In this case, all the character strings stored in the character string storing table 11 are checked. As indicated by ① in (c) of Figure 1, the search range of the table at the check extends to (carrier < 1 > is empty). The reason for checking all the character strings is that a plurality of character strings of an identical kind might be written in the table 11. That is, character strings in which characters except the parts enclosed with < > are the same, for example, (station < 1 > is empty) and (station < 2 > is empty) are the character strings of the identical kind.

In Figure 1, No. 1 rule has a coincident character string existent in the character string storing table 11 and the condition of the IF part thereof is met. Therefore, a character string (assign work < 3 >) which corresponds to the character string of the THEN part expressive of a conclusion is written into the character string storing table 11 anew (arrow ②).

Herein, in order to prevent the identical conclusion from being written into a plurality of areas of the character string storing table, the conclusion is written after it is acknowledged with a search range ② in (c) that the same is not included in the character strings already existing within the character string storing table 11. Regarding rules No. 2 and No. 3, the same processing is repeated to successively store the conclusions of THEN parts into the character string storing table 11. Since rule No. 2 includes a plurality of IF's, the conclusion of the THEN part is supplementarily written upon acknowledging that the AND logic of them holds.

Now, a method of coding a character string and a method of storing coded information on the basis of pointer coupling will be stated in order to elucidate the principle of the present invention.

(a) and (b) in Figure 2 are diagrams showing the format of the character string of an IF – THEN rule and the forms of IF – THEN rules, respectively.

As shown in (a) of Figure 2, a character string which expresses a condition and conclusion to be described in an IF – THEN rule is denoted as a character string enclosed with ( ). The character string is formed of fixed parts 21 which are fixedly handled, and parameter parts 22 which are handled as constants or variables. Each parameter part 22 is enclosed with < >. As shown in (b) of Figure 2, the IF part of the rule appoints after IF a character string of the format illustrated in (a) of Figure 2. In case of specifying a plurality of conditions, a plurality of IF (character string)'s are written in succession. The plurality of specified conditions are handled as AND logic. In (b) of Figure 2, ... denotes any desired character string, and signifies that a plurality of strings can be specified.

The THEN part of the rule appoints after THEN a character string of the format illustrated in (a) of Figure 2. In the IF – THEN rule indicated as Form 3 in (b) of Figure 2, a character string is appointed following a procedure name after THEN. In case of describing a plurality of conclusions, a plurality of THEN (character string)'s are written in succession.

IF – THEN rules include three sorts of forms shown in (b) of Figure 2, and these will now be described in order.

(i) Rule of Form 1: Rule in which only constants of facility Nos., job names etc. are described in parameter parts. In this form, the pattern matching with a character string in the work table is performed in all the parts (both the fixed and parameter parts) of the character string of the IF part. A character string to be supplementarily written in the work table is a character string as appointed in the THEN part.

(ii) Rule of Form 2: Rule which includes variables (W, X, Y, Z etc.) in parameter parts. A parameter part of a constant may well exist. In this form, the pattern matching with a character string in the work table is performed in those parts in the character string of the IF part which are other than the parameter parts specifying the variables. Next, in all the matched character strings within the work table, the contents of parameter parts corresponding to the parameter parts specifying the variables in the character string of the IF part are temporatily stored as the values of variables. A character string to be supplementarily written in the work table is such that the parameter parts specifying the variables in the character string of the THEN part are replaced with only those of stored variable values which meet the AND logic of the plurality of conditions of the IF part.

(iii) Rule of Form 3: Rule in which, besides the rule of Form 2, a procedure name is specified and enclosed with < > between the THEN and character string of the THEN part. Basically, the rule of this form is subjected to the same processing as that of the rule of Form 2. In the rule of this form, immediately before the character string of the THEN part is supplementarily written in the work table, control shifts to the specified procedure. In this case, the variable value which meets the AND logic of the plurality of conditions of the IF part is delivered as a subtrahend. In the procedure, such processing as maximum value selection and numerical value operations is performed, and the control can be returned by changing the delivered variable value. A character string to be supplementarily written in the work table is such that the variable value changed by the procedure is incorporated in the corresponding parameter part of the THEN part.

Figure 3 is an explanatory diagram of a method of turning a character string into an internal code.

In the character string which is appointed in the IF part or THEN part of the IF – THEN rule shown in Figure 2 and the character string which is stored in the work table, the fixed part 21 is merely an information for distinguishing a certain character string and another character string in point of processing. In contrast, as to the parameter part 22, the content itself becomes an information for deciding the operating command of the facility. For these reasons, the internal code is formed by a method in which the information items of the parameter parts 22 of the character string are handled as they are, and the fixed parts 21 are caused to correspond to one numeric value.

(a) in Figure 3 shows the format of the original character string, while (b) in Figure 3 shows the format of the internal code. ..... indicates any desired characters, 31 a numeric code, and 32 a parameter part.

In the system of Figure 3, first of all, the fixed parts 21 of the original character string are extracted, and one numeric code 31 is caused to correspond to each character string pattern. Accordingly, when the character string patterns of the fixed parts 21 are the same, the identical numeric code 31 is had. Next, after the numeric code 31, the parameter parts 22 of the original character string are arrayed in the order of appearance. These are called the parameter part 32. Owing to such internal code forming method, the processing of the pattern matching may judge the coincidence of only the parameter part 32 as to the character string of the same numeric code 31, and it can shorten the processing time as compared with the case of judging the coincidence of the whole character string as in Figure 1.

Figure 4 is a diagram showing examples of IF – THEN rules turned into internal codes.

In Figure 4, the internal codes of the IF – THEN rules in (a) of Figure 1 are exemplified. Here, (work is the head of a queue) is coded into numeric code 1, (assign work) into numeric code 2, (station is empty) into numeric code 3, (job of work) into numeric code 4, (station is usable) into numeric code 5, (station is the job alternative of work) into numeric code 6, and (operate on the less number station) into numeric code 7.

Figure 5A is an explanatory diagram illustrative of a method of storing the internal codes into the work table or a method of storing coded information on the basis of pointer coupling. In addition, Figure 5B is a diagram showing the pattern matching method and check ranges of the work table in correspondence with Figure 1.

As shown in Figure 5A, in the present system, only the parameter parts 32 in the coded information are taken out and are stored into the work table 50. In (a) of Figure 5A, the internal code information items of the rules shown in Figure 4 are extracted and arrayed. The numeric codes in the internal codes of the IF parts are x, while the numeric codes in the internal codes of the THEN parts are y.

As shown in (b) of Figure 5A, only the parameter parts 32 of the internal codes in (a) of Figure 5A except the numeric codes x and y are stored into the work table 50. In this case, the parameter parts 32 having the same numeric codes are coupled by pointers 51 and 52 and then stored in such a manner that which of the numeric codes x and y the stored parameter parts 32 correspond to can be distinguished. That is, the head pointer 51 is provided for each of the different numeric codes x and y, and the corresponding parameter parts 32 are stored ahead of the pointer 51. The parameter parts 32 having the same numeric codes are stored in the state in which they are coupled by the continuing pointers 52. The last of the continuing parameter parts 32 is indicated by the fact that the value of the continuing pointer 52 is "0".

Owing to this method, in the processing on the coincidence between the coded information of the IF part of the IF – THEN rule and the work table information, namely, the matching processing on the IF part, it is

possible to first trace the head pointer 51 corresponding to the numeric code (x, y) specified in the IF part and to sequentially process the parameter parts 32 stored in the work table 50.

Accordingly, in Figure 1, the search for the processing is conducted from the head of the work table 50, whereas in Figure 5B, all the useless checks can be omitted. As apparent by comparing (c) of Figure 1 and (c) of Figure 5B, the search range of the table 50 is much smaller in the latter than in the former. Likewise, as to the checks of the existence of the same information in the case of supplementarily writing the coded information of the THEN part into the work table 50, only the required minimum checks can be rendered sufficient in such a way that the pointers 52 are sequentially followed from the head pointer 51 which corresponds to the numeric code specified in the THEN part.

The present invention realizes a condition discriminative control on the basis of Figures 5A and 5B. It has the function of converting the statuses of the facilities into the internal codes and storing them into the work table in accordance with the foregoing method, the function of deciding control commands on the basis of the rules turned into the internal codes and the information coupled by the pointers and stored in the work table, and the function of converting the decided control commands into signals and sending the signals to the respective facilities.

Figure 6 is a block arrangement diagram of a facilities control apparatus showing an embodiment of the present invention.

The control apparatus is constructed of a memory part 601, a control part 602, facilities 603, status signal lines 604 and command signal lines 605. The memory part 601 has a plurality of information memory parts, namely, the areas of an input/output signal versus internal code information memory part 606, an IF – THEN rule trigger event memory part 607, an IF – THEN rule memory part 608, an internal code memory part 609, and a procedure information memory part 610. The memory part 601 is a part which stores the data of IF – THEN rules turned into internal codes, the correspondence table of external signals and the internal codes, etc. The control part 602 functions on the basis of the information stored in the memory part 601 to receive status signals indicative of the statuses of the facilities 603 through the signal lines 604, to decide control commands and to send control signals to the respective facilities 603 through the signal lines 605.

Figure 7 is a diagram for explaining the details of the I/O signal – internal code information memory part 606 in Figure 6. Figure 7, (a) shows a correspondence table for input signals and the internal codes, while Figure 7, (b) shows a correspondence table for the internal codes and output signals.

This memory part 606 is a part which stores information for the interface between the internal processing and the external signals of the status signal lines 604, the command signal lines 605, etc. shown in Figure 6.

The input signal – internal code correspondence table 71 shown in (a) of Figure 7 is a table which stores information for converting the signals of the status signal lines 604 into the internal codes to be handled in the control part 602. The internal codes are the codes explained with reference to Figure 3.

This table 71 is constructed of a status signal line entry area which stores the entry addresses of the status signal lines 604, a numeric code area which stores the internal codes expressive of facility statuses corresponding to cases where signals have developed on the signal lines 604, and a parameter part area. On the basis of the data of this table 71, the control part 602 accesses the respective status signal lines 604 and prepares the corresponding internal codes when the signals are generated. The prepared internal codes are stored into the internal code memory part 609. Here, in a case where the signals of the status signal lines 604 are binary information of "on" and "off", the control part 602 derives the internal codes stored in the table 71, in the form left intact, whereby the internal codes are deemed to have been prepared. For example, the internal code of address B in (a) of Figure 7 indicates such a case. In a case where the information items of the status signal lines 604 are numeric information such as the read information of bar codes, the internal codes to be prepared are such that the numeric information items are written as values into those parameter parts of the internal codes stored in the table in which the variables (such as W, X, Y and Z). For example, the internal code of address A in (a) of Figure 7 indicates such a case. More specifically, in a case where numeric information 34 develops at the status signal line address A, the control part 602 prepares an internal code "1 < 34 >". The internal code "1 < 34 >" indicates, for example, that work No. 34 has appeared at the head of a work providing line in an automated factory.

Referring now to (b) of Figure 7, the internal code – output signal correspondence table 72 is a table which stores information for sending control signals to the facilities 603 through the command signal lines 605 on the basis of internal codes expressive of the control commands decided by the control part 602. The internal codes expressive of the control commands are stored in the internal code memory part 609.

The table 72 is constructed of a numeric code area which stores the internal codes expressive of the control commands, a parameter part area, and a command signal line entry area which stores the entry addresses of the command signal lines 605 to send the control commands. The control part 602 compares the internal code expressive of the decided control command with the internal codes stored in this table 72 and delivers a signal to the command signal line 605 corresponding to the coincident internal code. In that case, when the variable (such as W, X, Y or Z) is not specified in the parameter part in the internal code stored in the table 72, the binary signal of "on" or "off" is outputted to the corresponding signal line address. As the case of providing the binary signal, there is address D in (b) of Figure 7 by way of example. On the other hand, when the variable is specified, the value of the parameter corresponding to the variable in the internal code expressive of the decided control command is delivered to the signal line address. For example, when the internal code expressive of the decided control command is "9 < 1 > < 34 > < 1 >", signal line address

C is supplied with values $\dot{3}\dot{4}$ and $\dot{1}$ in this order. The internal code "9 < 1 > < $\dot{3}\dot{4}$ > < $\dot{1}$ >" signifies, for example, the control command which instructs a carrier 1 in the automated factory to transfer work No. $\dot{3}\dot{4}$ to station $\dot{1}$.

Figure 8 is a detailed explanatory diagram of the IF – THEN rule trigger event memory part 607 in Figure 6. Figure 8, (a) shows a rule set trigger condition table, while Figure 8, (b) shows a storage area for No. of a rule set to be used.

The memory part 607 is a part which stores information for appointing a series of rules to be triggered in correspondence with the statuses of the facilities having arisen. Here, "to be triggered" is intended to mean "to be used when the control part 602 decides the control commands". For example, in the control of the automated factory, when a work has appeared at the head of the work providing line, a rule set for assigning the work to a work station is triggered, and a transfer command for a carrier and a job command for the work station are decided.

In addition, when a job has been completed in a certain work station, a rule set for carrying out a product is triggered, and a fabricated-article carry-out command for a carrier is decided. The rule set trigger condition table 81 is a table which stores information for appointing rule sets to be triggered corresponding to the respective statuses of the facilities 603 developed, as stated before. This table 81 is constructed of a status signal line entry area which stores the entry addresses of the status signal lines 604 corresponding to the arising facility statuses to be monitored, and a trigger rule set No. area which appoints Nos. of rule sets to be triggered when signals have developed on the signal lines 604. The used rule set No. storage area 82 is an area which stores No. of a rule set to be used for the control command decision by the control part 602. The control part 602 is polling the respective status signal lines 604 on the basis of the data of the rule set trigger condition table 81. Upon the occurrence of the signal, trigger rule set No. corresponding to the particular signal line 604 is stored into the used rule set No. storage area 82, whereupon the next processing is shifted to.

Figure 9 is a detailed explanatory diagram of the IF – THEN rule memory part 608 in Figure 6. Figure 9, (a) shows a rule set directory, while Figure 9, (b) shows a rule table.

The memory part 608 is a part which stores information on the IF – THEN rules descriptive of control logics. Since the present invention handles the IF – THEN rules in the form of internal codes, the IF – THEN rules turned into the internal codes are stored in the memory part 608. The rules to be stored are sorted into respective rule sets such as a rule set for assigning a work to a work station and a rule set for carrying out a product.

The rule set directory 91 is constructed of a rule set head position area which appoints the head positions of the respective rule sets on the rule table 92 storing the IF – THEN rules, and a rule set last position area which appoints the last positions thereof. For each rule set, the rules are stored between the head position and the last position. Nos. of the respective rule sets correspond to Nos. of the rows of the rule set directory 91.

In the rule table 92, each rule has the IF part and the THEN part divided, and the internal codes thereof are respectively stored in an IF part area and a THEN part area. Each area can store a plurality of internal codes. Further, the IF part area consists of a numeric code area for storing the numeric codes 31 of the internal codes and a parameter part area for storing the parameter parts 32 thereof. The THEN part area includes a numeric code area for storing the numeric codes 31 of the internal codes and a parameter part area for storing the parameter parts 32 thereof, and besides, a procedure name area for storing procedure names in the IF – THEN rules of Form 3. Here, "rule Type" is an information for distinguishing a rule for finally deciding the control command, from the other rules, and it is specified in a rule Type area. The rule to finally decide the control command is given "1", while the other rules are given "0". These information items are used for discerning which of conclusions expresses the control command to be finally obtained, when the control part 602 successively forms the conclusions (internal codes) of the THEN parts in the processing of deciding the control command.

Figure 10 is a detailed explanatory diagram of the internal code memory part 609 in Figure 6. (a) of Figure 10 shows a head pointer table, a parameter part storing table and a continuing pointer table; (b) thereof a control command storing table; (c) thereof a variable value temporarily storing table; and (d) thereof an AND logic meeting variable value-storing table.

This memory part 609 is a part for storing such information as the conclusions (internal codes) of the THEN parts which are successively formed, and the control command (internal code) which is finally decided, when the control command is decided on the basis of the internal codes expressive of the statuses of the facilities received and the IF – THEN rules, and the values of variables which are received when the IF – THEN rules of Forms 2 and 3 including the variables are processed. The parameter part storing table 101, the head pointer table 102 and the continuing pointer table 103 are work tables for storing by the method shown in Figure 5A the conclusions which are sequentially formed in the processing of the internal codes expressive of the statuses of the facilities externally received and the IF – THEN rules (among the internal codes appointed in the THEN parts, those except the control command finally decided). The parameter part storing table 101 is a table which stores the parameter parts 32 of the respective internal codes. The continuing pointer table 103 is a table which stores the continuing pointers 52 that couple the parameter parts 32 of the internal codes having the same numeric codes 31. The rows of the continuing pointer table 103 corresponding to those of the parameter part storing table 101 store the positions of those rows of the

parameter part storing table 101 in which the parameter parts 32 of the next continuing internal codes are stored. The last of the continuing parameter parts 32 is indicated by storing zero in the continuing pointer table 103. In addition, the head pointer table 102 is a table which stores the head pointers 51. The head pointer 51 indicates the head of the parameter parts 32 of the internal codes having the same numeric codes 31 as coupled by the continuing pointers 52. The rows of the head pointer table 102 correspond to the respective numeric codes 31. That is, numeric code 1 is at the first row, numeric code 2 is at the second row, and numeric code 3 is at the third row. The position of the head row of one series of parameter parts 32 having the same numeric codes 31 as stored in the parameter part storing table 101 is stored in the head pointer table 102. The head row of the empty areas of the parameter part storing table 101 is stored in an empty area head pointer area 104, all the empty areas are coupled by pointers, and the pointer information is stored in the continuing pointer table 103, so as to manage empty area positions.

The control command storing table 105 shown in (b) of Figure 10 is a table which stores the internal codes expressive of the control commands finally decided by the control part 602. This table 105 consists of a numeric code area which stores the numeric codes of the internal codes of the decided control commands, and a parameter part area which stores the parameter parts thereof. The variable value temporary storage table 106 in (c) of Figure 10 and the AND logic meeting variable value storage table 107 in (d) thereof serve to process the IF – THEN rules of Forms 2 and 3. When the judgement has been made on the coincidence between any desired internal code specified in the IF part of a rule and the internal codes stored in the work tables (head pointer table 102, parameter part storage table 101, continuing pointer table 103), the variable value temporary storage table 106 stores the parameter value of the parameter part 32 of that one of the internal codes stored in the work tables which coincides with the internal code specified in the IF part of the rule, the aforementioned parameter value being received as the value of the corresponding variable of the parameter part 32 of the internal code of the IF part of the rule. Whether or not any of the internal codes stored in the work tables coincides with the internal code specified in the IF part of the rule is judged by the use of the numeric code 31 of the internal code and the parameter part 32 thereof except parameters specifying variables. The AND logic meeting variable value storage table 107 is a table which stores those of the received variable values meeting the AND logic of the IF part.

Figure 11 is a detailed explanatory diagram of the procedure memory part 610 in Figure 6.

This memory part 610 is a part for storing the bodies and entry addresses of the procedures which are nominated in the IF – THEN rules of Form 3. A procedure entry table 111 consists of a procedure name area which stores the procedure names, and a procedure entry area which stores the entry addresses of the procedures. The respective procedures 112 are stored in entry address positions appointed in the procedure entry table 111.

Figure 12 is a functional block diagram of the control part 602 in Figure 6.

The control part 602 is constructed of a rule set trigger condition judging part 121, a facility status internal code generating part 122, a control command deciding part 123, and a control command signal generating part 124. In the flow of information, numeral 1201 designates a status signal, numeral 1202 rule set No., numeral 1203 the internal code of a facility status, numeral 1204 the internal code of a control command, and numeral 1205 a command signal.

Figure 13 is an operating flow chart of the control part 602.

The operations of the control part 602 will now be described in accordance with the flow of Figure 13, reference being had to the arrangements of Figures 7 to 12.

When the control part 602 has started operating, the polling of the status signal lines 604 is first initiated by the rule set trigger condition judging part 121. That is, the rule set trigger condition judging part 121 successively accesses the signal lines of the status signal line entries stored in the rule set trigger condition table 81, to check if a signal is generated or not (box 131). If any signal is not generated, the access of the successive signal lines 604 is cyclically continued. If any signal is generated, the trigger rule set number 1202 specified in the rule set trigger condition table 81 in correspondence with the particular signal line 604 is set in the area 82 for storing No. of a rule set to be used, and the facility status internal code generating part 122 is started (box 132). The facility status internal code generating part 122 first clears the head pointer table 102 to zero and sets the value of the empty area head pointer area 104 at "1". That is, the pointer is set at the head of the parameter part storage table 101. Further, the continuing pointer table 103 is set so that all the rows of the parameter part storage table may be coupled from the first row by the pointers. In other words, the status is established in which no internal code information is stored in the work tables (head pointer table 102, parameter part storage table 101, continuing pointer table 103). Subsequently, the signal lines of the status signal line entries stored in the input signal – internal code correspondence table 71 are successively accessed; the internal code (numeric code and parameter part) is generated for the signal line generating a signal, the internal code being specified in correspondence with the particular signal line in the input signal – internal code correspondence table 71 (refer to Figure 7); the internal code is stored into the work tables; and the control command deciding part 123 is started (box 133). Here, the storage of the generated internal code into the work tables is executed as follows. First, the pointer of the row of the head pointer table 102 corresponding to the numeric code 31 of the generated internal code is traced, and whether or not the parameter part coincident with that 32 of the generated internal code is stored in the parameter part storage table 101 is checked by tracing the pointers of the continuing pointer table 103. When the coincident parameter part is already existent, no action is taken. When it is not existent, that row of the parameter part

storage table 101 which the last of the traced pointers points (the continuing pointer before becoming zero points) is coupled to the row of the head of the empty areas. That is, the value of the empty area head pointer area 104 is set in that row of the continuing pointer table 103 which corresponds to the particular row. Next, the parameter part 32 of the generated internal code is stored into the row of the head of the empty areas, and the particular row is rendered the last row. In other words, zero is set in that row of the continuing pointer table 103 which corresponds to the particular row. In addition, the empty area head pointer area 104 is set at the row of the next empty area. The control command deciding part 123 decides an internal code representative of the control command by a process to be described later and stores it into the control command storage table 105, whereupon the control command signal generating part 124 is started (box 134). Unless the statuses of the facilities satisfy the condition of the control logic described by the IF – THEN rule, nothing is stored into the control command storage table 105. The control command signal generating part 124 first checks whether or not the internal code of the control command is stored in the control command storage table 105.

When nothing is stored, the rule set trigger condition judging part 121 is started without any processing, to start the polling again (box 131). When the control command internal code is stored, it is compared with the internal code stored in the internal code – output signal correspondence table 72, and the signal is generated for the signal line of the corresponding command signal line entry of the coincident row (refer to Figure 7). After the above signal generation processing has been performed for all the internal codes stored in the control command storage table 105, the rule set trigger condition judging part 121 is started (box 135), and the polling is started again (box 131).

Figure 14 is a diagram which shows the relationships of functional blocks and tables within the control command deciding part 123 in Figure 12.

In Figure 14, solid lines indicate the flows of controls, and thick lines the flows of information. The control command deciding part 123 consists of a control part 141 for deciding the control command, a condition judging part 142, a part 143 for generating an AND logic combination, an intermediate conclusion writing part 145, and a control command writing part 146.

Figure 15 is an operating flow chart of the control command deciding part 123 in Figure 14.

When the control command deciding part 123 has been started, the control is first delivered to the control command decision controlling part 141. Using the other functional blocks, the control command decision controlling part 141 decides the control command as stated below. The control command decision controlling part 141 first clears the control command storing table 105 (box 151), and on the basis of rule set No. stored in the used rule set No. storing area 82, it obtains from the rule set directory 91 the head and last positions in the rule table 92 for the rule to be used (box 152). The subsequent processing is executed for only the rule between the head and last positions.

Until any new rule no longer requires to supplement the internal code of the THEN part into the work tables, the control command decision controlling part 141 repeats the following processing for each of all the rules of rule Type 0 in the rule table 92 (box 1514). When there is no new rule requiring the supplement, the following processing is further repeated for each of all the rules of Type 1 in the rule table 92 (box 1515), whereupon the control command signal generating part 124 is started. The control command decision controlling part 141 takes out a certain rule, and performs processing on each internal code specified in the IF part of the rule. The condition judging part 142 is first called, and it judges the coincidence between one of the internal codes of the IF part of a certain rule and an internal code stored in the set of the work tables (box 153). The condition judging part 142 first traces the pointer of the row of the head pointer table 102 corresponding to the numeric code 31 of the internal code of the IF part and then traces the pointers of the continuing pointer table 103 in succession until the continuing pointer value becomes zero. Thus, it checks whether or not the parameter part coincident with the parameter part 32 of the internal code of the IF part is stored in the parameter part storing table 101. The judgement on the coincidence is done by the use of the parameter part 32 of the internal code of the IF part except the parameter parts specifying variables. In case of the coincidence, the value of the parameter of the parameter part in the parameter part storage table 101 corresponding to the parameter specifying the variable is stored into the variable value temporary storage table 106 as the value of the variable specified in the IF part. In this case, each time the parameter part coincident with that 32 of the internal code of the IF part is found in the parameter part storage table 101, the variable value is stored, and such parameter parts are successively stored in the respective rows of the variable value temporary storage table 106. In the case (rule of Form 1) where no variable is specified in the parameter of the parameter part 32 of the internal code of the IF part, no variable value is especially stored. In a case where, as the result of the above, no parameter part is coincident, the control command decision controlling part 141 extracts the next rule and begins the condition judgement again (box 154). In a case where any parameter part is coincident; if the parameter part 32 of the internal code of the IF part having now been judged on the coincidence includes a parameter specifying a variable, the AND logic combination generating part 143 is called and generates the combination of variable values meeting the AND logic of the IF part in conformity with a process to be described later (box 156). The result of the combination is stored in the AND logic meeting variable value-storing table 107. On this occasion, when there is no combination of the variable values meeting the AND logic, the control command decision controlling part 141 discards the rule being now processed, and it extracts the next rule so as to begin the condition judgement again (box 157). If the parameter part 32 of the internal code of the IF part having now been judged on the coincidence

includes no parameter specifying a variable, the above processing is not executed (box 155). After the control command decision controlling part 141 has performed the processing thus far described for all the internal codes specified in the IF part of the certain rule, it shifts to the next processing (box 158). First, the control command decision controlling part 141 extracts one internal code of the THEN part of the rule being now processed. In a case where a procedure name is nominated for the internal code in the rule table 92, the control command decision controlling part 141 calls a procedure triggering part 144 so as to trigger a nominated procedure (box 1510). In a case where no procedure name is nominated, the procedure triggering part 144 is not called (box 159). The procedure triggering part 144 searches the entry address of the procedure of the procedure name specified for the internal code, from the procedure entry table 111, and calls the procedure of the address with the AND logic meeting variable value-storing table 107 as a minuend. Within the procedure, the value of the AND logic meeting variable value-storing table 107 can be altered at will. Subsequently, the control command decision controlling part 141 examines the type of the rule being now processed (rule type in the rule table 92). In case of Type 0, the intermediate conclusion writing part 145 is called, and the extracted internal code of the THEN part of the rule being now processed is stored into the set of the work tables (box 1511). In case of Type 1, the control command writing part 146 is called, and the extracted internal code of the THEN part of the rule being now processed is stored into the control command storing table 105 (box 1512). The control command decision controlling part 141 repeats the processing thus far described for all the internal codes specified in the THEN part (box 1513). In a case where the parameter part 32 in the internal code intended to be stored includes a parameter specifying a variable, the intermediate conclusion writing part 145 creates an internal code in which the content of the parameter is replaced with the corresponding variable value of a certain row stored in the AND logic meeting variable value-storing table 107. On the other hand, in a case where the parameter part 32 includes no parameter specifying a variable, the internal code intended to be stored is used as it is. Subsequently, the intermediate conclusion writing part 145 stores the created internal code into the set of the work tables. The storage is done by the same method as in the foregoing case where the internal code generated by the facility status internal code generating part 122 is stored into the set of the work tables. On this occasion, as in the foregoing method, when the internal code intended to be stored is already stored in the work tables, it is not stored anew. The above processing is repeated for all the rows of the AND logic meeting variable value-storing table 107 in which variable values are stored. Likewise to the intermediate conclusion writing part 145, the control command writing part 146 first creates an internal code to be written. Subsequently, it stores the internal code into the control command storing table 105 successively from the head thereof. In this case, if the internal code to be stored is already existent in the control command storing table 105, it is not written anew. As in the case of the intermediate conclusion writing part 145, the above processing is repeated for all the rows of the AND logic meeting variable value-storing table 107 in which the variable values are stored.

Figure 16 is an operating flow chart of the AND logic combination generating part 143 in Figure 14.

The AND logic combination generating part 143 combines the combination (stored in the variable value temporarily storing table 106) of the values of the variables received when the coincidence between the certain internal code of the IF part and the internal code stored in the work tables has been judged anew and the combination (stored in the AND logic meeting variable value-storing table 107) of the values of the variables meeting the AND logic in the internal codes of the IF parts which have the coincidence judged before, to prepare anew the combination of the values of variables meeting AND logic when the internal codes subjected to the coincidence judgements are also included (the prepared combination is stored into the AND logic meeting variable value-storing table 107 again). More specifically, as shown in Figure 16, whether or not AND logic is met is judged for all the combinations between the rows of the variable value temporarily storing table 106 and the rows of the AND logic meeting variable value-storing table 107 (box 161). When the AND logic is met, the combination of variable values meeting the AND logic is created anew and is temporarily stored (box 162). Lastly, the new combination of the variable values stored temporarily is stored into the AND logic meeting variable value-storing table 107 again (box 163). Here, the case where the AND logic is met is a case where, on the side of the variable value temporarily storing table 106 and the side of the AND logic meeting variable value-storing table 107, the variables of an identical variable name have a coincident value, or either of the variables has no value (that is, it is blank). The combination of the variable values meeting also the new AND logic is created in such a way that, in one combination meeting the AND logic between the row of the variable value temporarily storing table 106 and the row of the AND logic meeting variable value-storing table 107, the variables having a coincident variable value are set at the coincident value, while as to the variables either of which is blank, the blank variable is changed into the value of the variable having this value.

Figure 17 is a diagram showing an example of creation of the combination of variables meeting AND logic. (a) in Figure 17 shows the variable value temporarily storing table, and (b) and (c) in Figure 17 show the AND logic meeting variable value-storing table.

In this example, the variable values (W = a, X = blank, Y = b, Z = blank) of the first row of the variable value temporarily storing table and the variable values (W = blank, X = e, Y = b, Z = blank) of the first row of the AND logic meeting variable value-storing table meet AND logic, so that the combination of the variable values meeting the AND logic, W = a, X = e, Y = b and Z = blank, is created in the first row of the AND logic meeting variable value-storing table anew.

In this way, according to the present embodiment, various control logics can be simply described using IF – THEN rules, and information turned into internal codes on the basis of them is merely stored into the respective tables of the memory part 601, whereby the condition discriminative controls of the various control logics can be easily performed. Further, the alterations of the contents of the condition discriminative controls attendant upon the change of a controlled system, the change of a control system, etc. can be readily coped with by altering the IF – THEN rules. Further, the present embodiment can shorten a period of time required for the pattern matching of character strings and a useless checking period of time attendant upon work table searches. Besides, by disposing the rule set trigger condition table 81, it is permitted to observe only the statuses of facilities requiring rule trigger, and the useless polling of signal lines can be reduced. For these reasons, the processing time since the occurrence of a certain facility status till the decision of a control command and the generation of the signal thereof is shortened as compared with that in the prior art, and the invention is applicable even to the control of a controlled system of which a quick control response is required. The operation of the control command deciding part 123 in the present embodiment was simulated. As a result, the processing rate was enhanced 20 times or more over the prior art, and the performance of several tens – hundred msec/decision could be fulfilled.

In the present embodiment, instead of polling the status signal lines by means of the control part 602, an external interrupt may well be used as the trigger of a series of operations of the control part 602. In this case, the control system of the present invention can be realized in such a way that an interrupt factor is assigned to each facility status which is the condition of rule trigger and that No. of a rule set to be triggered is appointed for each factor. Besides, when interrupt levels for the interrupt factors are specified in the rule set directory 91, priority levels can be given to the decisions of control commands. Moreover, when some rows of the head pointer table 102 are so set that the facility status internal code generating part 122 is not cleared to zero at the beginning of the control command decision, some internal codes can be fixedly stored in the set of work tables beforehand. This dispenses with the labor and time for receiving fixed conditions in the facilities from the signal lines on each occasion, so that the processing rate can be more enhanced.

As described above, according to the present invention, the character strings of IF – THEN rules and character strings to be stored in a work table are handled in the form of codes. In addition, in case of storing the coded information into the work table, the storage positions of information concerning the character strings of the same kind are coupled by pointers. Therefore, the checks of the information in the work table in the cases of judging the condition of an IF part and writing the conclusion of a THEN part may be performed in only necessary places, which permits to shorten the period of time required for the pattern matching of the character strings and to reduce the useless checks of the information in the work table, so that the processing speed can be sharply enhanced over the prior art. Accordingly, a specialist for developing a program is unnecessary even for the condition discriminative control of a controlled system requiring quickness, and a control technician can readily perform a desired condition discriminative control by describing a control system with IF – THEN rules by himself/herself.

CLAIMS

1. In a facilities control method having a rule memory part which stores rules consisting of conditions and conclusions of facilities, a status memory part which stores statuses of the facilities, job contents to be executed and contents concluded by the rules, and a rule applying part which compares the information stored in the status memory part and the conditions stored in the rule memory part and which stores the conclusions of the rules having had the conditions met into the status memory part, thereby to decide control commands for the facilities;

a facilities control method characterized by comprising control means to perform a control of receiving facility status signals expressive of the statuses of the facilities and delivering command signals responsive to the status signals to the facilities, first memory means to store information of corespondence between the status signals as well as the command signals and internal codes for internal processing by said control means, second memory means to store information for specifying a series of rules to be triggered in accordance with the statuses of the facilities having occurred, third memory means to store the rules turned into internal codes, and fourth memory means to store internal codes of conclusions sequentially formed in the process of deciding the control commands for the corresponding facilities on the basis of the internal codes expressive of the statuses of the facilities having occurred and the internal codes expressive of the rules, as well as internal codes finally decided,

said control means performing first processing of monitoring the facility status signals and setting, upon the generation of the signal, No. of a rule set for use on the basis of the information stored in said second memory means, second processing of generating the corresponding internal code from the generated facility status signal on the basis of the correspondence information stored in said first memory means, third processing of reading out the corresponding rules from said third memory means on the basis of No. of the rule set having been set by the first processing and deciding the final control command by referring to the information stored in said fourth memory means, and fourth processing of generating the command signal for the facility from the internal code of the decided control command on the basis of the correspondence information stored in said first memory means.

2. A facilities control method according to Claim 1, wherein said fourth memory means comprises a first table which stores parameter parts of the internal codes expressive of the statuses of the facilities having occurred and the internal codes of the conclusions formed according to the rules, a second table which stores continuing pointers for coupling on said first table the parameter parts of the aforementioned internal codes having the same fixed parts, and a third table which stores head pointers for indicating the heads of the series of parameter parts coupled by the continuing pointers for the respective same fixed parts.

3. A facilities control method substantially as herebefore described with reference to Figures 3 to 17 of the accompanying drawings.

Printed in the UK for HMSO, D8818935, 9/84, 7102.
Published by The Patent Office, 25 Southampton Buildings, London, WC2A 1AY, from which copies may be obtained.