A METHOD OF PATTERN RECOGNITION USING REWRITING RULES

Akira Kurematsu, Manabu Takeda, Seiichi Inoue
Research Laboratory, Kokusai Denshin Denwa Co. , Ltd.
Tokyo, Japan

Abstract

This paper presents a method of applying rewriting rules to the recognition of patterns, structure of which are sequentially connected and can be expressed as symbol strings. Rewriting rules, which play the part of a recognizer, are formulated in a hierarchical structure. Rule 1 normalizes the length of string and extracts properties, rule 2 transforms the deformed string in a context dependent way and rule 3 determines the category. A procedure for producing rules from given sample patterns based on the similarity between the symbols and a procedure for modifying the rules when input patterns are added are described. Experimental results of the recognition of Japanese spoken digits to illustrate the method are exhibited.

Introduction

Many efforts have been made to recognize complicated patterns such as connected speech, handwritten characters and image figures by the use of a machine. When abilities are compared, however, a machine is not up to a human being in recognizing patterns.

In most cases, patterns are recognized by the following method. The predetermined characteristic values of a given input pattern are calculated, and with a combination of the results, the output is given to one of the pre-established categories. However, as pointed out by Evans(1), it is difficult to improve the degradation in the performance with such a simple method when patterns are complicated, because patterns are recognized at one level. Moreover, it is necessary to obtain more information about pattern structure in a form better suited for further processing. In order to overcome such difficulties, the approach taken is directed to a hierarchical description of the structure of patterns in terms of subpatterns, their properties and the relations among them. Using this method, most of the initially designed recognizing machines can be utilized simply by analyzing the subpatterns deeply as required when the pattern is complicated. In addition, advanced pattern processing can be made easily by the use of a computer because an input pattern can be transformed to a symbol string and expressed in a form so that the machine can handle them.

As a simple form to use the syntax structure of patterns, sequential logic has been adopted to recognize spoken digits or handwritten characters and fairly good results have been attained. (2),(3) As indicated by Alter (4) and Fu(5), it is necessary to utilize contextual constraints and a knowledge of language strucsure to raise the recognition rate of continuous speech. As for the picture recognition, the idea of analyzing the structure of patterns is described initially in Minsky's "articular" description, (6) and many efforts have been made to research grammars and procedures for analysis suited to describe pictures. (7),(8).(9) However these have not led to a general theory especially in the stage to formulate grammars, although a few particular cases have been treated in some detail. The attempt to construct the general formalism intended to use in linguistic pattern analysis were outlined by Grenander(10) ,

This paper describes procedures for formulating rewriting rules for the recognition of patterns whose structure is one dimensional such as speech patterns, from symbol strings of sample patterns, based on the similarity between the symbols. A procedure for modifying the rules, when input patterns other than those used for the sample patterns are added, is also described. (11), (12) The above methoi was actually examined for the recognition of Japanese spoken digits, and this paper introduces the results.

Scheme of Pattern Recognition

The scheme of the pattern recognition which we deal with in this paper is shown in Fig. 1. An input pattern observed by observation measurements is segmented into subpatterns whose unit is properly determined. The subpattern is expressed by a vector in the measurement space. An input pattern can be expressed as a sequence of subpatterns. The subpattern is classified into one of predetermined classes of subpatterns, which are called fundamental constituents. By the use of symbols named to each fundamental constituent, the sequence of subpatterns is expressed as a string of symbols.

A symbol string of the pattern can be regarded as being generated by a set of the rewriting rules. That is, a set of symbol strings of patterns is regarded as a phrase structure language. In the language theory, it is shown that various classes of phrase structure languages are derived from a finite number of alphabet of symbols and rules, imposing restrictions on grammars, although they are not adequate for formulation of a full grammar for a natural language (13). In this paper, the discussions arc limited to the recognition of patterns such as limited spoken words, which are observed physically in the measurement space, therefore it can be assumed that the language is non-self-embedding and it is a finite state language. The content of a phrase in this case is a unit of a phonological forms which is sustained by the constrictions of the movements of human organs in making voice. These constrictions are expressed as rules.

A symbol string of the pattern is parsed by matching a sequence of a partial string by the use of rewriting rules. As a result of the parsing, a decision is made as to which category an input pattern belongs. For example, a speech pattern can be expressed as a sequence of frequency patterns observed at a certain interval. The frequency spectrum pattern is classified in accordance with the properties. The symbol may correspond to the name of phoneme.

In deriving the rewriting rules from the collection of sample patterns, the similarity between the symbols is considered to determine a group of symbols with common properties. The similarity between two subpatterns, X_1, X_2 is a scalar which represents a degree of likeness calculated in their measurement space. We denote it by $d(X_1, X_2)$.

The similarity between two symbols S_1, S_2 is defined by $d(S_1, S_2) \equiv d(X_{s1}, X_{s2})$, where X_{s1} and X_{s2} are vectors representing the classes (standard patterns) corresponding to S_1 and S_2 respectively.

We call a symbol attached to the fundamental constituent of a pattern a primitive symbol.

An intermediate symbol is named to a string of primitive symbols which share common properties and are considered to be a unit.

A symbol to express the category is called a category symbol.

We denote by S, Q, and P, respectively, primitive symbols, intermediate symbols and category symbols.

The structure of the rewriting rules ior the pattern recognition is formulated in a hierarchical structure. The hierarchical structure is effective in reducing the number of rules and in shortening the time required to refer to the rules. Depth of the hierarchy differs depending upon the complexity of the pattern. In this paper, we proceed with our discussion based on a model of three levels.

The application of the rules in the recognition of a symbol string is illustrated in Fig. 2. A primitive symbol string is transformed to an intermediate symbol string by rules of the 1st kind from left to right; the intermediate symbol string is transformed to another intermediate symbol string by rules of the 2nd kind from left to right; and, finally, the category is determined by rules of the 3rd kind.

The rule of the 1st kind (rule 1) deals with the extraction of properties from a primitive symbol string and the normalization of the length of a partial string, which, for example, is required for the recognition of speech spoken at different speeds.

The rule of the Znd kind (rule Z) transforms a partial string of intermediate symbols to another partial string. It deals with the normalization of the deformed patterns caused by various ways of production or the removal of the noise or the elimination of influences from neighboring subpattern It is expressed in a context dependent form

The rule of the 3rd kind (rule 3) determines the category from a string of intermediate symbols.

The rewriting rules can be formed in two structures;

(1) Structure in which rules for each category are separated,
(2) Structure in which rules for categories are combined.

In the separated structure, preparation and modification of the rules are easier although the number of rules increases, since rules are independent in the individual categories. In the combined structure, the number of rules can be reduced because rules existed together in two or more categories can be merged.

Procedure to Formulate Rewriting rules

The following procedure is described to formulate the rewriting rules by analyzing the structure of strings of given sample patterns.

An outline of the procedure for formulating the rules is described as follows: first the ordering relations among primitive symbols and intermediate symbols are examined, then a standard sequence for each category is derived, and then relations between partial strings are expressed in the form of rules.

Before describing the procedure, we will explain several terms used in it.

Stable symbol: A primitive symbol is called a stable symbol if it succeeds consecutively in more than a certain length (K_s). It corresponds to a stable portion in the pattern which will hardly be influenced with a noise.

Key symbol: A primitive symbol is called a key symbol if it occurs more than a certain times (K_k) in the same relative relations in the set of samples of the same category. It corresponds to the characteristics which are observed commonly in almost all patterns of the same category. The K_s and K_K are determined with regard to the nature of the pattern. An example is shown in Table 1 for the case that $K_s=2$ and $K_k=3$. Underlined symbol means the stable symbol.

Descendant: A primitive symbol is called a descendant of an intermediate symbol, say Q_i, if it is transformed to Q_i by rules of the 1st kind. A set of descendants of Q_i is denoted by $D(Qi) = \{ S_{il}, S_{i2} .... \}$ • A descendant is called a main descendant if it is a key symbol or if it occurs most frequently among the descendants of an intermediate symbol.

Standard sequence: A sequence of intermediate symbols corresponding to key symbols of a category is called a standard sequence of the category.

Ordering relation: An ordering relation among primitive symbols is an order of their occurrences in the set of primitive symbol strings of the same category.

Procedures to derive the rules from given sample patterns are described for two structures of rules.

Structure to separate the rules for each category

Step 1: Examine the ordering relations among primitive symbols for each category.

Step 2: Determine stable symbols and key symbols for each category. A procedure to decide key symbols from symbol strings of given sample patterns is described in the Appendix.

Step 3: Determine intermediate symbols and descendants for each category so that each primitive symbol may become a descendant of any intermediate symbol.

First an intermediate symbol is assigned to each key symbol. Next a primitive symbol which has not been picked up as a key symbol is assigned to a descendant in the following way. When there is a primitive symbol which is immediately in front of, rear of or in equal relation of the main descendant and the similarity to the main descendant is more than a certain value (K_d), such a symbol is put into the descendant of the intermediate symbol.

When one primitive symbol is a candidate of more than two descendants, such a symbol becomes a descendant of the intermediate symbol, the main descendant of which has the largest similarity value to the symbol. Those primitive symbols which are not assigned to any descendants are assigned to descendants of new intermediate symbols.

In the example shown in Table 1, intermediate symbols and descendants arc determined as shown in Table 2, when $d(S_1\ S_6) > K_d$, $d(S_1, S_4) > K_d$, $d(S_3, S_7) > K_d$, $d(S_8, S_6) > K_d$, $d(S_2, S_1) < K_d$, $d(S_2, S_0) < K_d$, $d(S_5, S_1) < K_d$, $d(S_5, S_0) < K_d$ and $d(S_2, S_5) > K_d$. It is to be noted that a new intermediate symbol Q_4 is assigned to S_2, because $d(S_2, S_1) < K_d$ and $d(S_2, S_0) < K_d$. Encircled symbols indicate main descendants.

Step 4: Form rules of the 1st kind based on the relations of intermediate symbols and descendants. When the length of a partial string to be transformed to the same intermediate symbol is not limited, rules for $D(Q_i) = \{ S_{i1}, S_{i2} \}$, for instance, are expressed as follows;

$$Q_i \rightarrow S_{i1},\ Q_i \rightarrow S_{i2},\ Q_i \rightarrow Q_i S_{i1},$$
$$Q_i \rightarrow Q_i S_{i2}.$$

A set of these rules is denoted by $Q_i \rightarrow S_{i1}, S_{i2};.$

Step 5: Transform a primitive symbol string to an intermediate symbol string by applying rules of the 1st kind and derive a standard sequence. Form rules of the 3rd kind to transform a standard sequence to a category symbol. Table 3 shows intermediate symbol strings and a standard sequence of sample patterns shown in the Table 1.

Step 6: Form rules of the 2nd kind by comparing a partial string of a standard sequence with a partial intermediate symbol string of a sample pattern. When there is a partial string which does not coincide with that of a standard sequence, it is expressed in the rule. The rules formed to recognize sample patterns in the Table 1 are shown in Table 4.

Structure to combine rules for all categories

In the structure to combine rules for all categories, intermediate symbols and descendants are to be merged because the rules are to be applied deterministically.

After the procedure proceeds from step 1 to step 3, intermediate symbols and descendants decided separately by the individual categories are combined and merged as follows.

Step 4': Form a table indicating all intermediate symbols and their descendants for individual categories. Merge and simplify the assignment of intermediate symbols and their descendants so that each primitive symbol becomes only one descendant. The procedure to merge the assignment is performed so that rules should be formed as simply as possible. But details are not presented in this paper.

Thereafter, the process goes to step 4 in the same way as in the case of the separated structure described above. After the completion of step 6, rules of the 2nd and 3rd kinds for all categories are combined.

Modification and Simplification of Rule

Modification of rule

Rules are to be modified in the following cases:

(i) The case that there is any sample pattern which is not recognized correctly by the rules after the completion of the step 6 in the preceding section.

(ii) The case that, after all given sample patterns are ascertained to be recognized correctly, other patterns, the category names of which are known, are not recognized correctly.

The rules should be modified so that patterns recognized correctly before the modification should be recognized once again correctly after the modification. The procedure for modifying the rules with the knowledge of the old rules, the similarity between the symbols and the ordering relations of the symbols in each category, is examined. The rules are modified in the following circumstances. Five cases where rules should be modified are enumerated;

(a) The first case is that an input pattern is recognized as belonging to more than two categories; we call such a grammar ambiguous. This case occurs when the rules are separated into each category. In this case there are at least two symbols which commonlv occur in at least two categories. Those symbols are included commonly with other symbols in a set of descendants of a particular intermediate symbol in at least one of those categories.

For example, assume that two symbols, S_k and S_m, are included commonly in two categories i and j, and the following rules exist,

$$Q_{ik} \rightarrow S_k;,\ Q_{im} \rightarrow S_m;,\ Q_{jk} \rightarrow S_k, S_m;,$$
$$P_i \rightarrow Q_{ik} Q_{im},\ P_j \rightarrow Q_{jk},$$

where Q_{ik} and Q_{im} are intermediate symbols in category i and Q_{ik} is in category j. In this case a symbol string, $W = S_k S_m$, is transformed to QikQim in the category i and transformed to Qjk in the category j, which causes ambiguity. A grammar is revised so that it becomes unambiguous by adding a new intermediate symbol to the category where descendants are included commonly. In the example above, a new intermediate symbol, Q_{jm} i» added to the category j, and rules of the 1st kind are modified as

$Q_{jm} \rightarrow S_m$;, $Q_{jK} \rightarrow S_K$;. Rules of the 2nd kind are added in response to the change of rules of the 1st kind.

(b) In the second case, a pattern is rejected because primitive symbols are not transformed to intermediate symbols by rules of the 1st kind. Primitive symbols not transformed are those which did not occur in the sample patterns used to construct the rules. The modification of the rules is carried out for rules of the 1st kind. The best assignment of primitive symbols to descendants is determined by the use of the values of the similarity between those primitive symbols and descendants.

(c) In the third case, an input pattern is rejected because the intermediate symbol string is not transformed by any rules of the 3rd kind. This occurs due to the lack of rules of the 2nd kind to transform a partial string of intermediate symbols of an input pattern to a standard sequence of the corresponding category. For this problem, there are two solutions; rules of the 2nd kind are added or rules of the 1st kind are changed so that descendants of intermediate symbols are replaced. In the latter case, however, rules of the 2nd kind may also have to be modified in response to the change in rules of the 1st kind. In either way, an increase in the number of the rules caused by the modification is examined, and whichever has less number of rules is adopted.

(d) In the fourth case, an input pattern is misrecognized because a set of symbol strings to be recognized as belonging to some category is so large that even a symbol string which is to be recognized as belonging to another category is included. This trouble occurs because there are so many descendants of some intermediate symbols that some string is recognized as in more than two categories. In this case, rules of the 1st and 2nd kinds are modified so that a set of symbol strings to be recognized for each category may become adequate.

(e) In the fifth case, an input pattern is not recognized correctly because rules of the 2nd kind which should be applied to the string of another category are applied. This trouble occurs when rules of all categories are combined, and the length of the context of rules of the 2nd kind is insufficient. In this case, the problem can be solved by increasing the length of the context as required. An example is introduced as follows:

It is assumed that there is a set of rules expressed as follows in the rules of the 2nd kind:

(a) $Q_i Q'_k Q_j \longrightarrow Q_i Q_k Q_j$

(b) $Q_k Q'_j Q_m \longrightarrow Q_k Q_j Q_m$

Now, when the rules (a) and (b) are to be applied in this order for an intermediate symbol string of an input pattern expressed as $W = Q_i Q_k Q_j Q_m$, the W becomes QiQkQjQm by the application of the rule (a). However, when the recognition cannot be made correctly unless the W becomes Q^Q^QjQj^ using the rule (b), the length of the rule (b) must be increased to $QiQkQjQ_m \rightarrow QiQ_kQjQ_m$.

<u>Simplification of rule</u>

It is necessary to simplify the rules so that the number of rules is reduced by remaining rules which are essential to the recognition and deleting those which do not contribute to the recognition. There are two ways to simplify the rules. In one way, the number of rules of the 2nd kind is reduced by reducing that of intermediate symbols. By this method, those primitive symbols precisely determined at the beginning are merged. In the other way, the length of a standard sequence of a category expressed by the right term of the rule of the 3rd kind is reduced so that portions required in classifying patterns among other categories are remained and those not essential to the contribution to the classification are eliminated. By this method, some rules of the 1st and 2nd kinds can be eliminated. Some patterns may be rejected even for the simple reason that a partial symbol string which does not contribute to the classification does not coincide with the same portion of the standard string, when the simplification is not conducted. When the simplification is adopted, they are correctly recognized.

Experiment

An experiment to recognize Japanese spoken digits was carried out to examine the method described above.

The reasons why this method is adopted for the recognition of limited vocabulary are that by this method modified patterns by the influences of coarticulations of neighboring phonems and differences of speeds in utterance can be transformed in a unified way, and moreover, connected speech can be recognized without noticing the segmentation of speech.

Data to be recognized were ten Japanese digits; /ICHI/, /NI/, /SAN/, /YON/, /GO/, /ROKU/, /NANA/, /HACHI/, /KYU/, and /REI/. Each digit was pronounced separately. The speech was analyzed and read into a computer by the frequency analysis computer input system illustrated in Figure 3. Rectified filter outputs were sampled at an interval of approximately 50 ms and converted to digital signals.

As a trial to extract fundamental patterns corresponding to primitive symbols from given patterns, the mode-seeking training method similar to Bonner(4) was applied to frequency spectrum patterns of 100 data spoken by 10 male persons, where the clusters were formed by comparing the distances of the frequency spectrum patterns.

First, by the comparison of outputs of the frequency analysis, the frequency spectrum pattern at each sampling point was classified into four major classes; vowel, voiced consonant, unvoiced consonant and stop.

Except for "stop", patterns of three classes were classified more precisely; vowels were classified into 24 classes, voiced consonants into 5 classes, and unvoiced consonants 4 classes. Primitive symbols were named to these classified patterns. Table 5 shows them. As the value of the similarity between primitive symbols, the Euclidean distance of the spectrum pattern was adopted. Using 100 previously used sample patterns, rules were derived by following the procedures described before for the case of the separated structure of rules, the structure of symbol strings was analyzed at each category; stable symbols and key symbols were derived, and intermediate symbols and their descendants were decided based on the similarities between primitive symbols. Table 6 shows primitive symbol string of class /SAN/. The spectrogram of the sound /SAN/ and the corresponding primitive symbols can be seen in Figure 4. Table 7 shows intermediate symbols and their descendants for class /SAN/ Encircled symbol means the main descendant.

Next, primitive symbols were transformed to intermediate symbols and the standard sequence was derived. Table 8 shows intermediate symbol strings and standard sequence of /SAN/. Subsequently, rules of the 2nd and 3rd kinds were constructed. Table 9 shows them. B is the symbol which indicates the beginning of a string. X is the symbol which is not transformed by rule 1. In the rule of the 2nd kind, rules of a special kind were included: They eliminate the symbol X although the symbol X is inserted between two intermediate symbols indicated in the standard sequence. It is effective in removing a noise of the pattern while recognizing unknown input patterns.

After completely forming the rules to classify all given sample patterns first, 100 more patterns pronounced by 10 other male persons were imposed. Seventy out of them were recognized correctly, two were misrecognized and others were rejected. The rules were modified so that these additional patterns would be recognized correctly in accordance with the procedures described previously, and thus, the rules with which all 200 patterns could be recognized correctly were successfully established.

The same experiment was conducted also for the combined structure of rules. Rules to recognize all given patterns were composed.

Table 10 shows comparison of the number of rules in each case. The number of rules to recognize the first 100 sample patterns is shown. That for 200 patterns including 100 additional patterns is shown in parenthesis.

Conclusion

A pattern recognizing method in reference to the rewriting rules of phrase structure grammar has been described. Patterns arc transformed into a symbol string, and recognized by transforming partial strings from left to right in accordance with the rules. The rules are formed in three hierarchy levels. Procedures to form and modify the rewriting rules by the use of knowledge of the symbol strings of given sample patterns and values of the similarity between symbols are described. The results of an experiment to recognize spoken Japanese digits are introduced.

It was shown that procedures described in this paper worked well for the recognition of actual patterns of spoken Japanese digits. It is thought that the number of rules to recognize ten spoken digits should be decreased. It is expected that the higher recognition rate will be accomplished by revising the primitive symbols and forming rules by collecting enough sample patterns.

In the pattern recognition by the adoption of rewriting rules, the most important factor which seriously affects result of recognition is how to decide primitive symbols, that is, how to pick up fundamental elements of the pattern. When primitive symbols which adequately reflect characteristics of pattern are determined, number of rules can be reduced and applicability of the rules can be increased.

So far as the primitive symbol string of one category is not identical to that of another, these two patterns can be classified as precisely as desired by increasing length of the context. This is one of the merit of this method. As for the ability of this method to recognize unknown patterns, high recognition rate can be attained by increasing the number of sample patterns so that sample patterns thoroughly represent statistical property of patte rns .

In this method, recognition cannot be made within a short period of time because rules are referred one by one. Time may be reduced by dividing rules into groups. Moreover, when a set of input symbol strings can be regarded as regular set, it is possible to implement *a.* machine by the use of sequential circuits and recognition can be made within real time.

When the separated structure is compared with the combined structure, number of rules for the combined structure is smaller. However, as long as recognition ability is concerned, the separated structure is more advantageous from the comparison of the recognition rate and number of rules for unknown input patterns. Moreover, as for time required in recognition, the separated structure excels the combined structure, if it is possible to process rule applying operation in parallel for all categories .

It is also possible to add rules of the 3rd kind instead of those of the Znd kind during modifying rules. This means that patterns of one category can be divided into two or more classes.

Pertaining to the selection of rules, no definite criterion used to determine at which kind amongst the 1st, Znd and 3rd kinds should the rules be modified has not yet been given, except for the increase of number of rules. However, another way is left to modify rules by using estimated recognition rate of the rule as it can be defined.

Correct recognition can be attained by giving priority to output of recognition result although the grammar is ambiguous.

The advantages of the method for recognizing the patterns by the use of rewriting rules in the form mentioned in this paper are that the description of the rewriting rules which express the structure of the pattern can be understood easily and intuitively and that the complex patterns will be recognized at a high recognition rat e.

Acknowledgements

The authors express their deep appreciation to Dr. S. Oshima and Dr. Y. Nakagome for their continual support and encouragement provided toward us for this study. The authors' thanks arc also expressed to their colleagues in the Information Processing Research Group of KDD Research Laboratory lor zealous discussions and stimulating valuable suggestions.

APPENDIX

Procedure to Decide Key Symbols

A procedure to derive key symbols from symbol strings of the sample patterns is introduced. A relative position (R) of a symbol in a symbol string is defined as $R\text{-}L_S/L$, where L is the length of a symbol string and L_S indicates the position of the symbol taken from the left end of the string. In this case, if two or more same symbols continue, the L_S is placed at the center of the succession. Let the occurrence (F) be the number of strings containing a particular symbol in the symbol strings of the sample patterns. When the symbol is separated in two or more positions in a string, the leftmost symbol is subjected for the examination of F. This method is called a leftmost principle.

Step 1 : The occurrence (F) for all stable symbols is calculated and the symbol for which the F is highest is detected. This symbol is referred to as S_M. A relative position of the S_M in the string containing S_M is denoted as $R(S_M)$, and the mean relative position is denoted as "$R(S_M)$, where such symbols as are identical with the S_M the relative position of which is within a fixed value (γ_1) are picked up. When the occurrence (F) of the picked up symbol is greater than a fixed value (θ_2), the symbol is a key symbol.

Step 2: In the symbol strings containing SM, a symbol which is located in the left of SM and the occurrence (F) of which is greater than a fixed value ($\theta_1 j$) is taken in accordance with the leftmost principle. When the occurrence (F) of the symbol is greater than a fixed value (θ_1), it is denoted as Sp. For a string having no SM a symbol identical with Sp the relative position R(Sp) of which is located within a range of a fixed value (γ_1) is taken. When the occurrence (F) of Sp thus counted is greater than a fixed value ($\theta 2$). it is a key symbol.

For symbols located to the right of the SM, the same procedure applies. This step proceeds until all symbols whose occurrences (F) are greater than a fixed value ($\theta 2$) are taken.

References

(1) T. G. Evans, "A Grammar-Controlled Pattern Analyzer", Proceedings of IFIP Congress 1968, pp. 1592-1598.

(2) P. N. Sholtz, R. Bakis, "Spoken Digit Recognition Using Vowel Consonant Segmentation", JASA, vol. 34, pp. 1-5, Jan. 1962.

(3) H. Genchi, K-I. Mori, S. Watanabe, S. Katsuragi, "Recognition of Handwritten Numerical Characters for Automatic Letter Sorting", Proc. of IEEE, vol. 56, No. 8, pp. 1292-1301, Aug. 1968.

(4) R. Alter, "Utilization of Contextual Constraints in Automatic Speech Recognition", IEEE Trans., vol. AU-16, No. 1, pp. 6-11, March 1968.

(5) Y. Fu, "Application of Automata Theory to Automatic Speech Recognition", Proc. of Hawaii International Conference on System Science (1968).

(6) M. Minsky, "Steps toward Artificial Intelligence", Proc. IRE., vol. 49, No. 1, Jan. 1961.

(7) R. Kirsch, "Computer Interpretation of English Text and Picture Patterns", IEEE Trans., vol. EC-13, No. 4, pp. 363-376, Aug. 1964.

(8) R. Narasimham, "Syntax-Directed Interpretation of Classes of Pictures", Comm. ACM, vol. 9, pp. 166-173, March 1966.

(9) R. S. Ledly, L. S. Rotolo, J. B. Wilson, M. Belson, T. J. Golab and J. Jacobson, "Pattern Recognition Studies in the Biomedical Sciences", 1966 Proc. SJCC, pp. 411-431, 1966.

(10) U. Grenander, "Linguistic Tendencies in Pattern Analysis", IBM Scientific Center Report (1968).

(11) S. Inoue, A. Kurematsu, "New Method of Recognition of Spoken Digits", 1968 National Conference of the Institute of Electronics and Communication Engineers of Japan, No. 59.

(12) A. Kurematsu, M. Takeda, S. Inoue, "Pattern Recognition by Use of the Generative Grammar of the Phrase Structure Language", Technical Report of the Research Group on Automata Theory of the Institute of Electronics and Communication Engineers of Japan, A69-70, Jan. 1969.

(13) N. Chomsky, "On Certain Properties of Grammars", Inform, and Control, vol. 2, pp. 137-167, 1959.

(14) R. E. Bonner, "A 'Logical Pattern' Recognition Program", IBM. J., vol. 6, pp. 353-360, 1962.

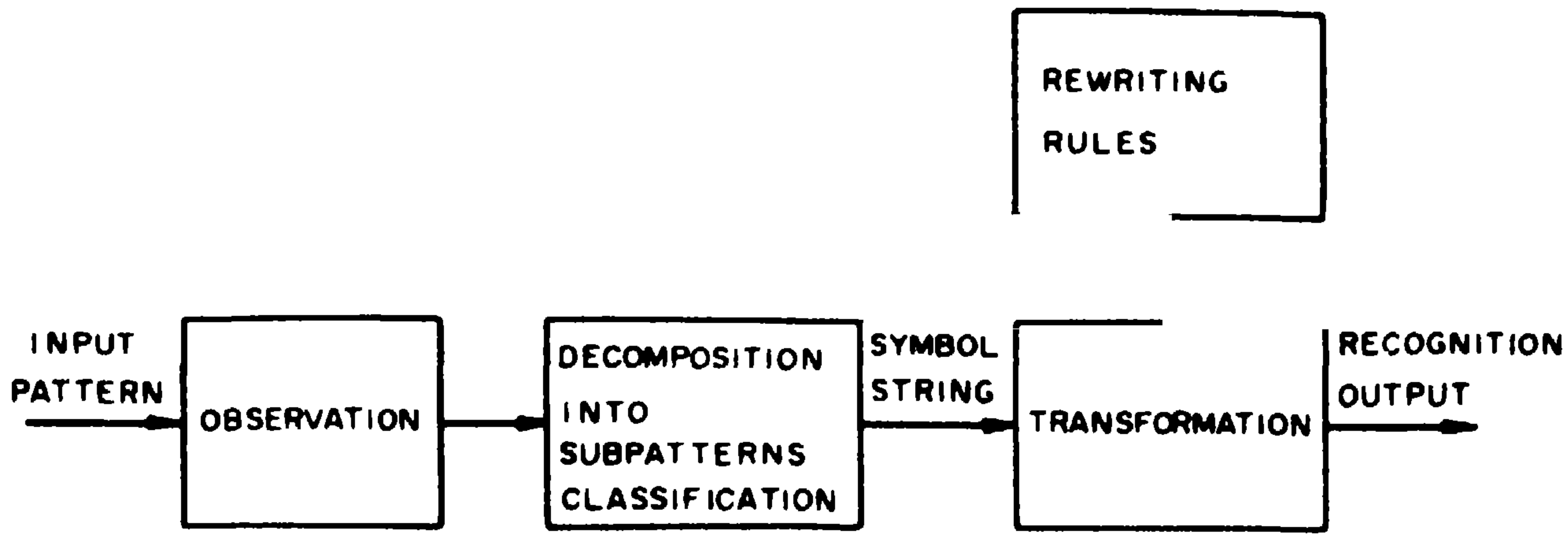

Fig.1 Scheme of pattern recognition

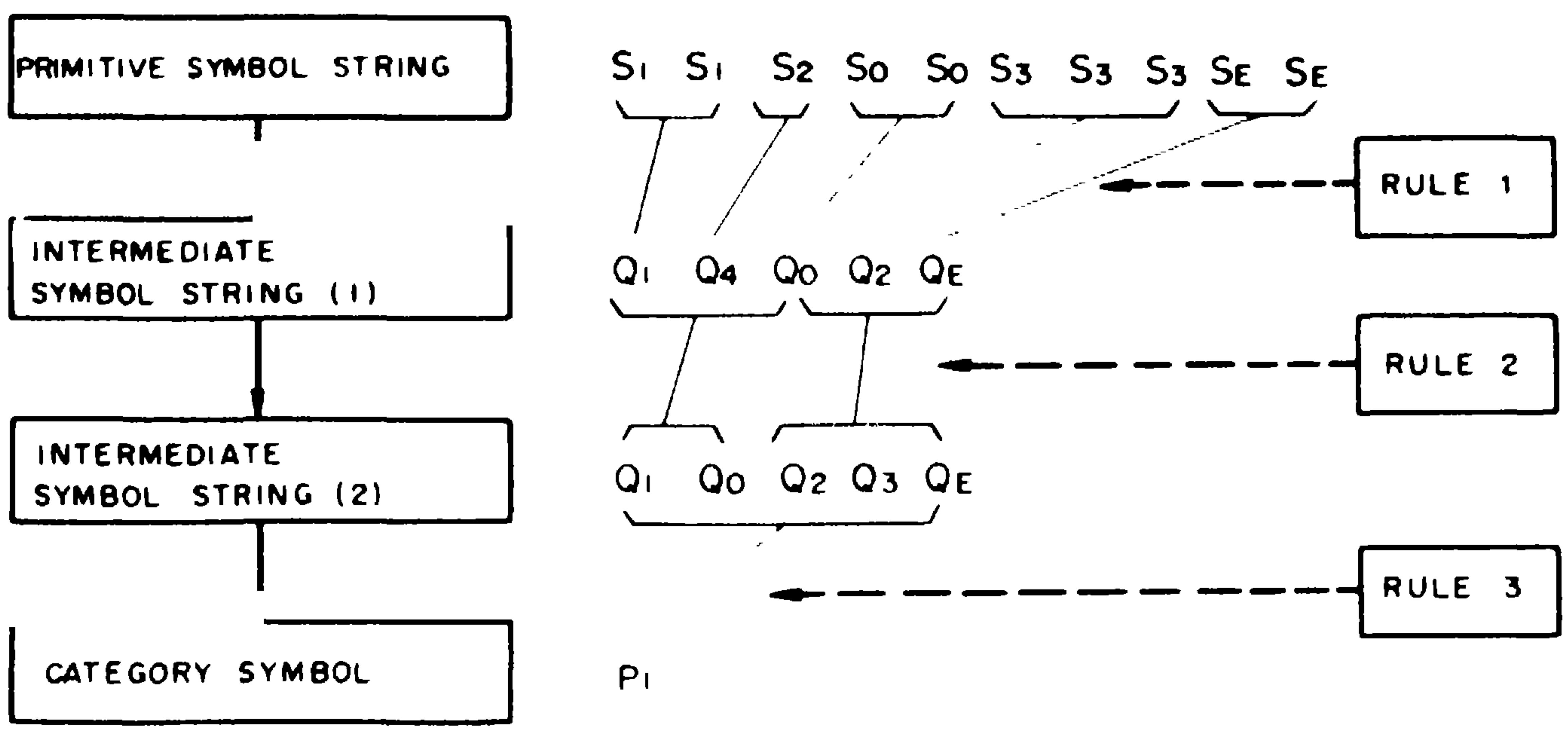

Fig 2 Application of rules in the recognition of symbol string

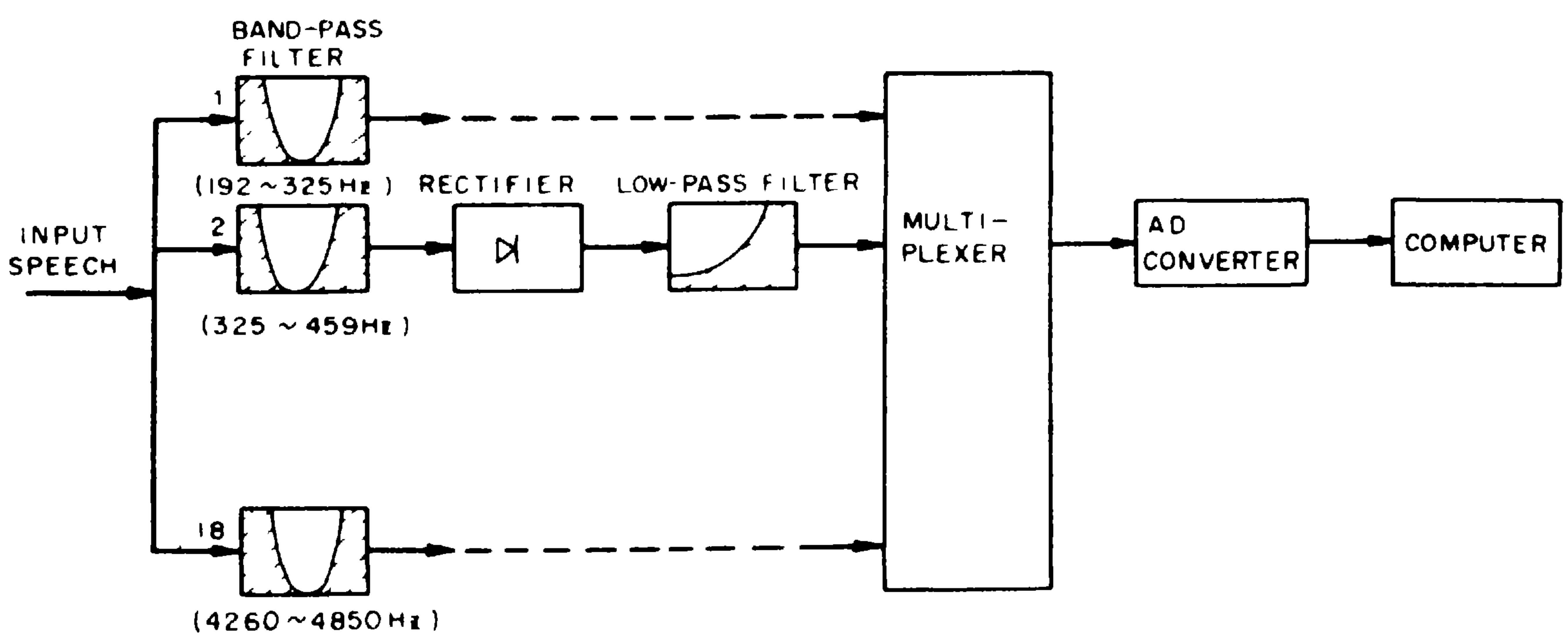

Fig 3 Frequency analysis and computer input system

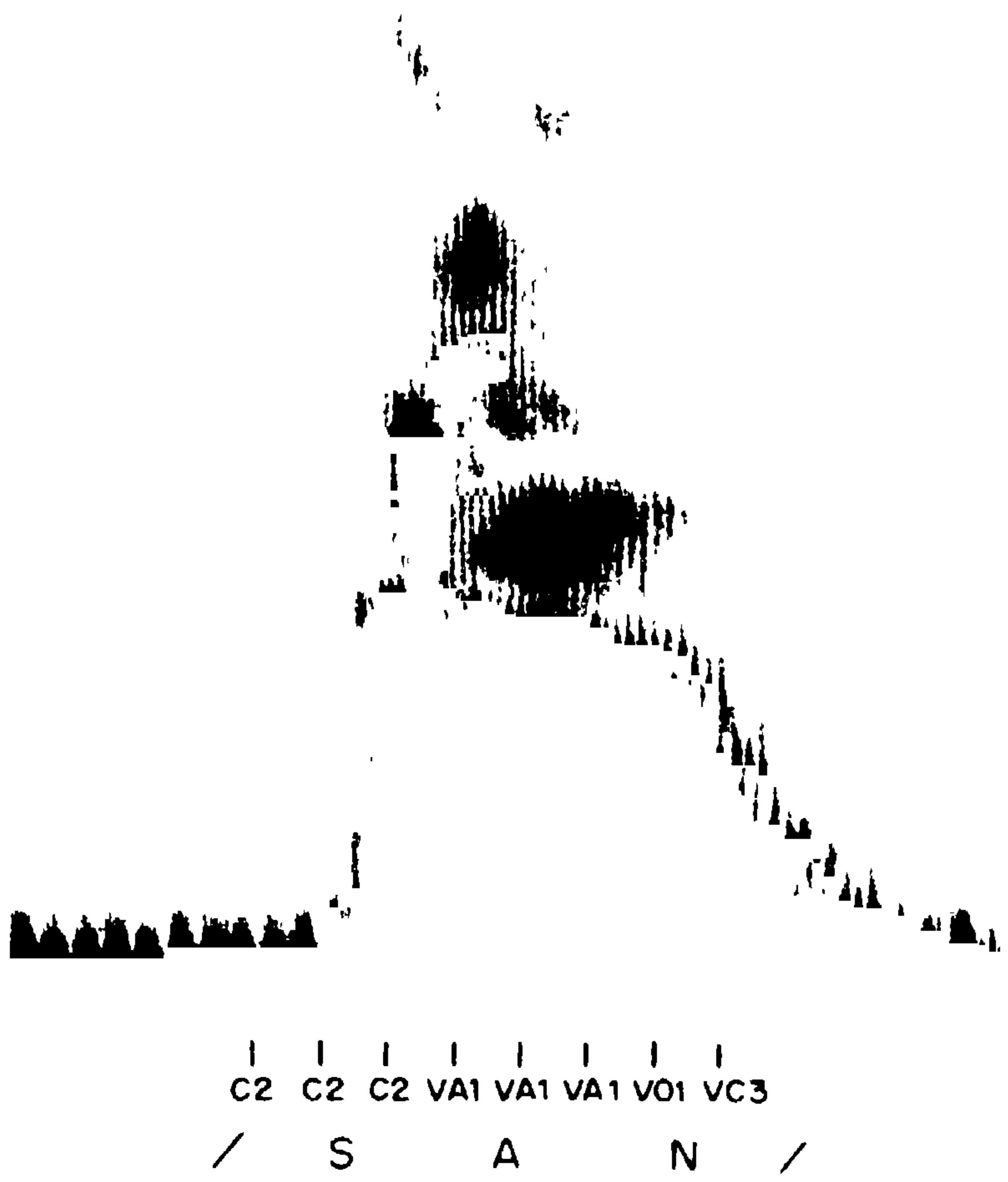

Fig.4 Spectrogram for Japanese spoken digit /SAN/

Table 1. Example to pick up stable symbols and key symbols.

	Primitive symbol string
W_1	S_1 S_1 S_2 S_0 S_0 S_3 S_3 S_3 S_E S_E
W_2	S_1 S_1 S_0 S_0 S_3 S_3 S_8 S_8 S_E S_E
W_3	S_4 S_5 S_0 S_0 S_3 S_3 S_8 S_8 S_8 S_E S_E
W_4	S_1 S_6 S_2 S_0 S_0 S_3 S_7 S_8 S_E S_E
Key symbol	S_1 S_0 S_3 S_8 S_E

Table 2. Intermediate symbol and descendant.

Intermediate symbol	Descendant
Q_0	(S_0)
Q_1	(S_1) S_6 S_4
Q_2	(S_3) S_7
Q_3	(S_8) S_6
Q_E	(S_E)
Q_4	(S_2) S_5

Table 3. Expression of intermediate symbol string.

	Intermediate symbol string
W_1	Q_1 Q_4 Q_0 Q_2 Q_E
W_2	Q_1 Q_0 Q_2 Q_3 Q_E
W_3	Q_1 Q_4 Q_0 Q_2 Q_3 Q_E
W_4	Q_1 Q_4 Q_0 Q_2 Q_3 Q_E
Standard string	Q_1 Q_0 Q_2 Q_3 Q_E

Table 4. List of rewriting rules.

Rule 1	$Q_1 \rightarrow S_1, S_6, S_4,$
	$Q_2 \rightarrow S_3, S_7,$
	$Q_4 \rightarrow S_2, S_5,$
	$Q_0 \rightarrow S_0$
	$Q_3 \rightarrow S_8, S_6,$
	$Q_E \rightarrow S_E;$
Rule 2	$Q_1\ Q_0 \rightarrow Q_1\ Q_4\ Q_0$
	$Q_2\ Q_3\ Q_E \rightarrow Q_2\ Q_E$
Rule 3	$P_1 \rightarrow Q_1\ Q_0\ Q_2\ Q_3\ Q_E$

Table 5. List of primitive symbol for the recognition of Japanese spoken digit.

Vowel	/A/	VA1, VA2, VA3, VA4
	/I/	VI1, VI2, VI3
	/U/	VU1
	/E/	VE1, VE2, VE3, VE4, VE5
	/O/	VO1, VO2, VO3, VO4, VO5, VO6, VO7, VO8, VO9, VO10, VO11
Voiced consonant		VC1, VC2, VC3, VC4, VC5
Unvoiced consonant		C1, C2, C3, C4

Table 6. Primitive symbol string for sample patterns of /SAN/.

No of samples	Primitive symbol string
3 - 1	VC3 VA1 VA1 VC2 VC2 V02 E
3 - 2	C2 C2 C2 VA1 VA1 VA1 VO1 VC3 E
3 - 3	C1 C2 C1 VA1 VA1 VA1 VA1 E
3 - 4	C1 C1 C2 VA1 VA1 VA1 VA1 VU1 E
3 - 5	C1 C1 C1 C1 V05 V01 V01 E
3 - 6	C2 VC5 VA4 VO1 VU1 VC1 E
3 - 7	C1 C2 C2 VE2 VA1 VA1 VO1 VO2 E
3 - 8	C1 C1 C1 VI3 VA2 VA1 VC3 VC2 VC2 E
3 - 9	C1 VA3 VA1 VA1 VE2 VC1 E
3 - 10	C1 C1 C1 VC2 VA1 VA1 VA1 VO1 VO2 VO2 E

Table 7. Intermediate symbols and those descendants of class /SAN/.

Intermediate symbol	Descendant
Q31	C1, C2
Q32	VC3
Q33	VC2, VF2
Q34	VA1, VA2, VA3, VA4, VI3, VE2, VC5
Q35	VO1, VU1, VE2, VO5
Q36	VC3
Q37	VC2
Q38	VO2, VC1

Table 8. Intermediate symbol string and standard symbol sequence of /SAN/.

No of samples	Intermediate symbol string
3 - 1	Q32 Q34 Q34 Q37 Q37 Q38 E
3 - 2	Q31 Q31 Q34 Q34 Q35 Q36 E
3 - 3	Q31 Q31 Q34 Q34 E
3 - 4	Q31 Q31 Q34 Q34 Q35 E
3 - 5	Q31 Q31 Q35 Q35 E
3 - 6	Q31 Q34 Q34 Q35 Q35 Q38 E
3 - 7	Q31 Q31 Q33 Q34 Q34 Q35 Q38 E
3 - 8	Q31 Q31 Q34 Q34 Q36 Q37 Q37 E
3 - 9	Q31 Q34 Q34 Q35 Q38 E
3 - 10	Q31 Q31 Q33 Q34 Q34 Q35 Q38 Q38 E
Standard sequence	Q31 Q34 Q35 Q38 E

Table 9. List of rule 2 and rule 3 for the recognition of /SAN/.

Rule 2	Q34 Q35 Q38 → Q34 Q37 Q37 Q38
	Q34 Q35 Q38 E → Q34 E
	Q35 Q38 E → Q35 E
	Q31 Q34 Q35 → Q31 Q35
	Q34 Q35 Q38 E → Q34 Q36 Q37 Q37 E
	Q31 Q34 → Q31 Q33 Q34
	B Q31 Q34 → B Q32 Q34
	B Q31 → B X Q31
	Q31 Q34 → Q31 X Q34
	Q34 Q35 → Q34 X Q35
Rule 3	P3 → Q31 Q34 Q35

Table 10. Comparison of number of rules.

		Rule 1	Rule 2	Rule 3	Total
Separated structure	/ICHI/	7 (7)	7 (7)	1 (1)	15 (15)
	/NI/	5 (5)	5 (7)	1 (1)	11 (11)
	/SAN/	8 (8)	10 (11)	1 (1)	19 (20)
	/YON/	7 (7)	7 (9)	1 (1)	15 (17)
	/GO/	6 (6)	7 (9)	1 (1)	14 (16)
	/ROKU/	13 (13)	7 (7)	1 (1)	21 (21)
	/NANA/	10 (10)	5 (8)	1 (1)	16 (19)
	/HACHI/	10 (10)	5 (5)	1 (1)	16 (16)
	/KYU/	7 (7)	6 (7)	1 (1)	14 (15)
	/REI/	5 (7)	9 (14)	1 (1)	15 (22)
	Total	78 (80)	68 (84)	10 (10)	156 (174)
Combined structure		11 (11)	82 (124)	17 (18)	110 (153)