

- [22] L. G. Lajtha and R. Oliver, "Cell population kinetics following different regimes of irradiation," *British J. Radiol.*, vol. 35, pp. 131–140, 1962.
- [23] C. L. Lawson and R. J. Hanson, *Solving Least Squares Problems*. Englewood Cliffs, NJ: Prentice-Hall, 1974.
- [24] J. M. Mendel, "Multistage least-squares parameter estimators," *IEEE Trans. Automat. Contr.*, vol. AC-20, pp. 775–782, Dec. 1975.
- [25] L. W. Nelson and E. Stear, "The simultaneous on-line estimation of parameters and states in linear systems," *IEEE Trans. Automat. Contr.*, vol. AC-21, pp. 94–98, Feb. 1976.
- [26] C. C. Paige and M. A. Saunders, "Least squares estimation of discrete linear dynamic systems using orthogonal transformations," *SIAM J. Numer. Anal.*, vol. 14, pp. 180–193, Apr. 1977.
- [27] L. Padulo and M. A. Arbib, *System Theory*. Philadelphia, PA: Saunders, 1974.
- [28] S. Perry, "Cell kinetics and cancer therapy: History, present status, and challenges," *Cancer Treatment Rep.*, vol. 60, pp. 1699–1704, Dec. 1976.
- [29] A. P. Sage and J. L. Melsa, *System Identification*. New York: Academic, 1971.
- [30] E. Tse and J. Anton, "On the identifiability of parameters," *IEEE Trans. Automat. Contr.*, vol. AC-17, pp. 637–645, Oct. 1972.
- [31] M. A. Van Dilla, T. T. Trujillo, P. F. Mullaney, and J. R. Coulter, "Cell microfluorometry: A method for rapid fluorescence measurement," *Science*, vol. 163, pp. 1213–1214, 1969.
- [32] C. Van Loan, "Lectures in least squares," Dep. Comput. Sci., Cornell Univ., *Tech. Rep.*, TR 76-279, 1976.
- [33] L. M. Van Putten, H. J. Keizer, and J. H. Mulder, "Perspectives in cancer research: Synchronization in tumor chemotherapy," *European J. Cancer*, vol. 12, pp. 79–85, 1976.

# Attributed Grammar—A Tool for Combining Syntactic and Statistical Approaches to Pattern Recognition

WEN-HSIANG TSAI, MEMBER, IEEE, AND KING-SUN FU, FELLOW, IEEE

**Abstract**—Attributed grammars are defined from the pattern recognition point of view and shown to be useful for descriptions of syntactic structures as well as semantic attributes in primitives, subpatterns, and patterns. A pattern analysis system using attributed grammars is proposed for pattern classification and description. This system extracts primitives and their attributes after preprocessing, performs syntax analysis of the resulting pattern representations, computes and extracts subpattern attributes for syntactically accepted patterns, and finally makes decisions according to the Bayes decision rule. Such a system uses a combination of syntactic and statistical pattern recognition techniques, as is demonstrated by illustrative examples and experimental results.

## I. INTRODUCTION

COMBINING syntactic and statistical pattern recognition approaches has been advocated by several investigators [1]–[4], [7], [22] in the past decade. The motivation arises from the fact that neither the syntactic

approach nor the statistical approach alone is adequate for some practical applications; the former is weak in handling noisy patterns and numerical semantic information [10], [24], and the latter is unable to describe complex pattern structures and subpattern relations. Since the advantage of one approach appears to be the disadvantage of the other, a hybrid model is desirable that incorporates the advantages of both and is thus more useful to real applications. In this paper we propose the use of attributed grammars as a tool for combining the syntactic and the statistical approaches to pattern recognition.

Attributed grammars were first formulated by Knuth [5] to assign semantics or meanings to context-free languages from the computational linguistics point of view. Each production rule of an attributed grammar consists of a syntactic rule and a semantic rule, the former being used to specify language syntax and the latter to add contextual semantics. Illustrative examples of applications of such semantic formalism to patterns described by picture description languages (PDL) [6] are found in [7]. The necessity of using pattern semantics to facilitate the utilization of contextual information was also emphasized in [23], although attributed grammars were not used there. Tang and Huang [8] applied attributed grammars to image understanding. You and Fu [9], [25] used attributed gram-

Manuscript received November 15, 1979; revised September 10, 1980. This work was supported by the Office of Naval Research under Contract N00014-79-C-0574.

W. H. Tsai was with the School of Electrical Engineering, Purdue University, West Lafayette, IN 47907. He is now with the Institute of Computer Engineering, National Chiao Tung University, Hsinchu, Taiwan, Republic of China.

K. S. Fu is with the School of Electrical Engineering, Purdue University, West Lafayette, IN 47907.

mers for shape recognition. All the above applications of attributed grammars are deterministic in nature without statistical consideration. In this study we try to inject statistical decision theory into the use of attributed grammars so that the resulting formalism might be more practical for applications.

Statistical consideration is possible due to the fact that primitive and subpattern attributes contained in the semantic rules of attributed grammars can be randomized to take care of noise and distortions. This means that probability distribution or density functions can be inferred to specify different occurrence possibilities of varying attributes. Optimum decisionmaking becomes possible through the use of these functions. In this paper we propose a pattern analysis system using attributed context-free grammars which, in addition to their pattern structure analysis and description capability using syntactic production rules, can also make statistical decisions for pattern classification according to the pattern attributes computed in the semantic attribute rules.

Without the guidance of structural analysis, numerical feature or attribute extraction sometimes may not be effective enough for pattern classification [3], [9]. In the proposed system, it is shown that semantic rules of the attributed grammars can be used to guide effective primitive or subpattern attribute extraction. This is made possible through the use of syntactic production rules and parsing procedures to group primitives hierarchically into subpatterns and patterns. Both local and global attribute extractions can be specified in the semantic rules in the form of mappings, functions, or computation algorithms. All useful attributes, either from primitives or from subpatterns can be put into a so-called total attribute vector for final decisionmaking.

## II. ATTRIBUTED GRAMMARS FOR PATTERN DESCRIPTION AND CLASSIFICATION

In this section, we give a definition of attributed grammars. An example to illustrate the usage of such grammars is provided. A pattern analysis system using attributed grammars is then described.

### A. Definition and Example of Attributed Grammars

*Definition 1:* An attributed context-free string grammar is a 4-tuple  $G = (V_N, V_T, P, S)$  where

- $V_N$  set of nonterminals,
- $V_T$  set of terminals,
- $S \in V_N$  start symbol,

for each  $X \in (V_N \cup V_T)$ , there exists a finite set of attributes  $A(X)$ , each attribute  $\alpha$  of  $A(X)$  having a set, either finite or infinite, of possible values  $D_\alpha$ ; and  $P$  is a set of productions each of which is divided into two parts: a syntactic rule and a semantic rule. The syntactic rule is of the following form

$$X_0 \rightarrow X_1 X_2 \cdots X_m$$

where  $X_0 \in V_N$  and each  $X_i \in V_N \cup V_T$  for  $1 \leq i \leq m$ . The semantic rule is a set of expressions of the following form

$$\begin{aligned} \alpha_1 &\rightarrow f_1(\alpha_{11}, \alpha_{12}, \cdots, \alpha_{1n_1}) \\ \alpha_2 &\rightarrow f_2(\alpha_{21}, \alpha_{22}, \cdots, \alpha_{2n_2}) \\ &\vdots \\ \alpha_n &\rightarrow f_n(\alpha_{n1}, \alpha_{n2}, \cdots, \alpha_{nn_n}), \end{aligned}$$

where  $\{\alpha_1, \alpha_2, \cdots, \alpha_n\} = A(X_0) \cup A(X_1) \cup \cdots \cup A(X_m)$ , each  $\alpha_{ij}$  ( $1 \leq i \leq n, 1 \leq j \leq n_i$ ) is an attribute of some  $X_k$  for  $0 \leq k \leq m$ , and each  $f_i$  ( $1 \leq i \leq n$ ) is an operator which may be in one of the following three forms:

- a) a mapping  $f_i: D_{\alpha_{i1}} \times D_{\alpha_{i2}} \times \cdots \times D_{\alpha_{in_i}} \rightarrow D_{\alpha_i}$ ,
- b) a closed-form function, i.e.,  $\alpha_i$  may be expressed functionally in terms of the values of  $\alpha_{i1}, \alpha_{i2}, \cdots, \alpha_{in_i}$ ,
- c) an algorithm which takes  $\alpha_{i1}, \alpha_{i2}, \cdots, \alpha_{in_i}$  and any other available information or data<sup>1</sup> as input and  $\alpha_i$  as output.

The above definition follows Knuth's formalism closely [5]; two kinds of attributes are included in the semantic rules: inherited attributes and synthesized attributes. The former are those aspects of meaning which come from the context of a phrase in a string, whereas the latter are those aspects of meaning which are built up from within the phrase. Note that a phrase represented grammatically by a nonterminal corresponds to a subpattern consisting of several primitives. In the above definition, if  $\{\alpha_1, \alpha_2, \cdots, \alpha_n\} = A(X_0)$  and each  $\alpha_{ij}$  is an attribute of some  $X_k$  for  $1 \leq k \leq m$  (not  $0 \leq k \leq m$ ), then all attributes defined in the semantic rules are *synthesized attributes*. If  $\{\alpha_1, \alpha_2, \cdots, \alpha_n\} = A(X_1) \cup A(X_2) \cup \cdots \cup A(X_m)$  (not including  $A(X_0)$ ) and each  $\alpha_{ij}$  is an attribute of  $X_0$ , then all attributes defined are *inherited attributes*.

In syntactic pattern recognition, if a top-down parsing is adopted to analyze pattern structures, then inherited attributes are more convenient for use because they can be computed in a top-down fashion, starting from the start symbol  $S$  of the grammar [26]. On the contrary, if a bottom-up parsing is preferred, then synthesized attributes should be used, which are computed in a bottom-up fashion. In the following, we will give an example of chromosome classification to illustrate the use of attributed grammars. Chromosomes will be described by a grammar with synthesized attributes, although a chromosome grammar with inherited attributes has also been proposed [27]. For illustrative purposes, the example is kept simple and involves only a single attribute. A more complicated example can be found in Tsai and Fu [30].

*Example 1: Chromosome Classification—An Illustrative Example:*

1) *Conventional syntactic approach:* With the following primitives as terminals [11],

$$V_{TM} = \{a, \curvearrowright; b, \curvearrowleft; c, \curvearrowup; d, \curvearrowdown\}$$

<sup>1</sup>With respect to pattern recognition, this means the whole input pattern.

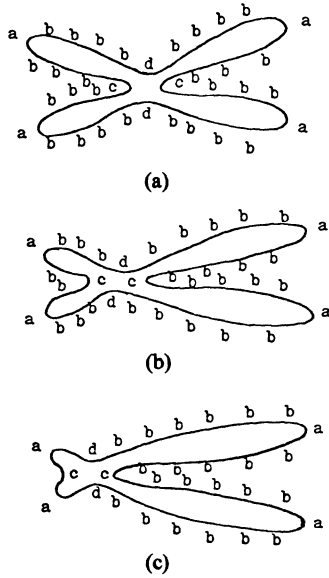


Fig. 1. Three classes of chromosomes. (a) Median chromosome  $z_M = cbbbabbbdbbbabbbcbbbbabbbdbbbabbb$ . (b) Submedian chromosome  $z_S = cbabbbdbbbabbbcbbbbabbbdbbbab$ . (c) Acrocentric chromosome  $z_A = cadbbbbbabbbcbbbbabbbdbbdac$ .

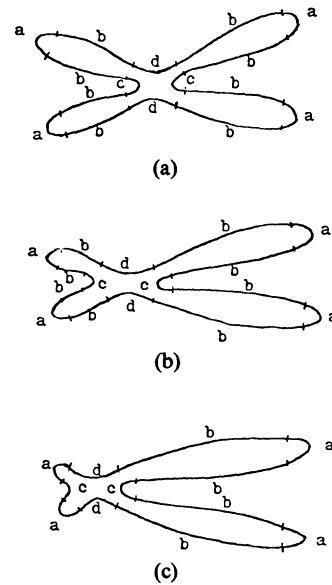


Fig. 2. Three classes of chromosomes. (a) Median chromosome  $z_M = dbabcabdbabcab$ . (b) Submedian chromosome  $z_S = dbabcabdbabcab$ . (c) Acrocentric chromosome  $z_A = dbabcabdaca$ .

three kinds of chromosomes—median, submedian, and acrocentric—are segmented accordingly, and are shown in Fig. 1 together with their string representations.<sup>2</sup>

The nonattributed grammars to characterize the three kinds of chromosome patterns could be as follows [12]:

$$\begin{aligned}
 G_{\text{median}} &= (V_{NM}, V_{TM}, P_M, S) \\
 V_{NM} &= \{S, A, B, D, H, J, E, F\}, V_{TM} = \text{as above} \\
 P_M: S &\rightarrow AA, D \rightarrow FDE, H \rightarrow a \\
 A &\rightarrow cB, D \rightarrow d, J \rightarrow a \\
 B &\rightarrow FBE, F \rightarrow b, E \rightarrow b \\
 B &\rightarrow HDJ
 \end{aligned}$$

$$\begin{aligned}
 G_{\text{submedian}} &= (V_{NS}, V_{TS}, P_S, S), V_{TS} = V_{TM} \\
 V_{NS} &= \{S, A, B, D, H, J, E, F, W, G, R, L, M, N\} \\
 P_S: S &\rightarrow AA, D \rightarrow FDE, G \rightarrow FG, L \rightarrow HNJ \\
 A &\rightarrow cM, D \rightarrow FG, W \rightarrow WE, R \rightarrow HNJ \\
 B &\rightarrow FBE, D \rightarrow WE, F \rightarrow b, G \rightarrow d \\
 B &\rightarrow FL, L \rightarrow FL, E \rightarrow b, W \rightarrow d \\
 B &\rightarrow RE, R \rightarrow RE, H \rightarrow a, N \rightarrow FDE \\
 M &\rightarrow FBE, J \rightarrow a
 \end{aligned}$$

$$\begin{aligned}
 G_{\text{acrocentric}} &= (V_{NA}, V_{TA}, P_A, S), V_{TA} = V_{TM} \\
 V_{NA} &= \{A, B, D, H, J, E, F, L, R, W, G\} \\
 P_A: S &\rightarrow AA, D \rightarrow FG, G \rightarrow FG, R \rightarrow HDJ \\
 A &\rightarrow cB, D \rightarrow WE, W \rightarrow WE, G \rightarrow d \\
 B &\rightarrow FL, L \rightarrow FL, L \rightarrow HDJ, W \rightarrow d \\
 B &\rightarrow RE, R \rightarrow RE, H \rightarrow a, E \rightarrow b \\
 J &\rightarrow a, F \rightarrow b.
 \end{aligned}$$

A close observation of the three kinds of chromosomes shows that they can be easily classified by measuring the lengths of their arm pairs [25]. The length of an arm pair is represented by the average of the lengths of the two arms on either side of a chromosome. If the length of the left arm pair is approximately equal to that of the right

<sup>2</sup>In the remainder of this paper, by a string representation we mean the string of terminal symbols (not including attributes).

arm pair, then the pattern is a median chromosome. If they differ significantly, then it is a submedian chromosome, or if the length of one arm pair is near zero, then it is an acrocentric chromosome. However, since conventional grammars such as those shown above cannot incorporate attributes (numerical data) within their production rules, the classification consequently can only rely on *symbolic* syntax analysis which, when made to take care of numerical information contained in the input patterns, is usually not very effective and efficient.

2) *Attributed grammar with synthesized attributes for chromosomes*: We now show that an attributed grammar with synthesized attributes can be used for chromosome description and classification. First, we remove the fixed-length restriction from terminal  $b$ , and let its total curve length be an attribute. For example in Fig. 1 (a) we consider the curve between terminal  $d$  and terminal  $a$  as a *single* terminal  $b$ . The resulting segmentations and string representations for the chromosomes are shown in Fig. 2. Note that the strings for a median and submedian chromosomes have become identical, but this is not a problem since our discrimination between them will rely on the difference of their synthesized attributes. The grammar is given as follows in which superscripts of nonterminals are used just for the purpose of discriminating identical nonterminals on the right-hand side of a syntactic rule.

A chromosome grammar with synthesized attributes:

$$\begin{aligned}
 G_S &= (V_T, V_N, P, S) \\
 V_T &= V_{TM} \text{ as specified previously,} \\
 V_N &= \{S, Q_1, Q_2, R_1, R_2, M_1, M_2, A, B, C, D\}
 \end{aligned}$$

with attribute sets as follows:

- a)  $A(A) = A(C) = A(D) = A(M_2) = \phi$ ,
- b)  $A(X) = \{l_X\}$ , for  $X = B, M_1, R_1, R_2, Q_1, Q_2$ ,
- c)  $A(S) = \{l_{S1}, l_{S2}\}$ .

- $P$ :
- 1)  $S \rightarrow Q_1^1 Q_2^2; l_{S1} \leftarrow l_{Q_1^1}, l_{S2} \leftarrow l_{Q_2^2}$
  - 2)  $S \rightarrow Q_1^2 Q_2^1; l_{S1} \leftarrow l_{Q_1^2}, l_{S2} \leftarrow l_{Q_2^1}$
  - 3)  $S \rightarrow Q_2^2 Q_1^1; l_{S1} \leftarrow l_{Q_2^2}, l_{S2} \leftarrow l_{Q_1^1}$

where nonterminal  $S$  represents the whole pattern,  $Q_1$  or  $Q_2$  represents an arm pair connected to a primitive  $d$ , attribute  $l_{S1}$  or  $l_{S2}$  is an arm pair length in the chromosome, and  $l_{Q_1}$  or  $l_{Q_2}$  is the average length of the two arms in the left or right arm pair:

- 4)  $Q_1 \rightarrow DR_1; l_{Q_1} \leftarrow l_{R_1}$
- 5)  $Q_2 \rightarrow DR_2; l_{Q_2} \leftarrow l_{R_2}$

where nonterminal  $R_1$  or  $R_2$  represents an arm pair,  $D$  represents the primitive  $d$ , and attribute  $l_{R_1}$  or  $l_{R_2}$  is the average length of the two arms in left or right arm pair.

- 6)  $R_1 \rightarrow M_1^1 CM_1^2; l_{R_1} \leftarrow (l_{M_1^1} + l_{M_1^2})/2$
- 7)  $R_2 \rightarrow M_2^2 CM_2^1; l_{R_2} \leftarrow 0$

where nonterminal  $M_1$  or  $M_2$  represents a chromosome arm,  $C$  represents the primitive  $c$ , and attribute  $l_{M_1}$  or  $l_{M_2}$  represents the length of arm  $M_1$  or  $M_2$ .

- 8)  $M_1 \rightarrow B^1 AB^2; l_{M_1} \leftarrow (l_{B^1} + l_{B^2})/2$
- 9)  $M_2 \rightarrow A$
- 10)  $A \rightarrow a$
- 11)  $B \rightarrow b; l_B \leftarrow l_b$
- 12)  $C \rightarrow c$
- 13)  $D \rightarrow d$

where nonterminals  $A, B, C, D$  represent primitives  $a, b, c, d$ , respectively and attribute  $l_B$  or  $l_b$  is the curve length of primitive  $b$ .

3) *Discussions*: Besides being a single grammar instead of three,  $G_S$ , though in context-free form, is a finite-state grammar. The introduction of attributes into a grammar, generally speaking, reduces grammatical complexity [9], [25]. This is indeed an advantage of using attributed grammars for pattern recognition because a reduction of grammatical complexity implies efficient syntax analysis.

To classify a given chromosome  $\omega$  using  $G_S$ , a bottom-up parsing is more convenient for structural analysis since only synthesized attributes are defined in  $G_S$ . After the string of the chromosome is parsed, two attributes,  $l_{S1}, l_{S2}$ , are synthesized for the start symbol  $S$  which represents the whole pattern. Such attributes will be expressed by a vector  $X_\omega$  which we will call the total attribute vector (TAV) of  $\omega$ . Here we have  $X_\omega = (l_{S1}, l_{S2})$ . Since chromosomes vary in sizes, for each class of chromosomes there exists a density function which specifies the distribution of all possible TAV's of this class. Using this function and other statistics (see the following sections), various statistical decision criteria for classification can be applied. Actually, the whole theory of statistical pattern recognition is applicable here. An example of statistical chromosome classification using grammar  $G_S$  will be given later in this paper. Although attributes are introduced mainly for the purpose of statistical classification, they

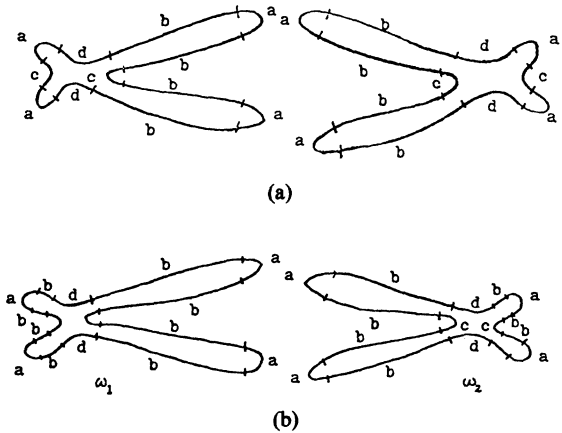


Fig. 3. Two acrocentric chromosomes in opposite directions. (a) With string representations  $z_2$  (left) and  $z_3$  (right). (b) With same string representation  $z_1$  ( $l_{S2}$  of  $\omega_1$  and  $l_{S1}$  of  $\omega_2 \approx 0$ ).

also contribute to the description of patterns. For details, see Tsai and Fu [27].

Finally, it should be noted that the chromosome grammar  $G_S$  can generate three syntactically different string representations by using different combinations of syntactic rules during string derivations:

$$\begin{aligned} z_1 &= dbabcbbabdbabcbab \\ z_2 &= dbabcbbabdaca \\ z_3 &= dacadbabcbab \end{aligned}$$

in which  $z_2$  and  $z_3$  both represent the structures of acrocentric chromosomes but in opposite directions (see Fig. 3(a)).  $z_1$  primarily represents a median or a submedian chromosome, but it can also represent an acrocentric chromosome because, as mentioned previously, we allow all  $b$  terminals to vary in length so that, even on the short arm pair of an acrocentric chromosome,  $b$  terminals with very small lengths may still exist after boundary segmentation (see Fig. 3(b)).

#### B. A Pattern Analysis System Using Attributed Grammars

The block diagram of a syntactic pattern recognition system using attributed grammars is shown in Fig. 4. It is modified from a similar diagram for conventional syntactic pattern recognition [7]. Given an input pattern for classification, after preprocessing, all necessary primitives and their attributes are extracted according to some pre-specified procedures. Note that only primitive attributes are extracted here; no subpattern attributes are to be extracted at this stage because which primitives are to be grouped to form a subpattern is still unknown. The next step is to transform the primitive set into a structural representation, such as a string, a tree, or a graph, by assigning symbols to primitives, selecting concatenating directions, and any other pre-specified relations. The resulting representation is then analyzed syntactically by using the syntactic rules of the attributed grammars, while the semantics computation is performed simultaneously to obtain all required nonterminal (subpattern) attributes according to the semantic rules. On the other hand, during

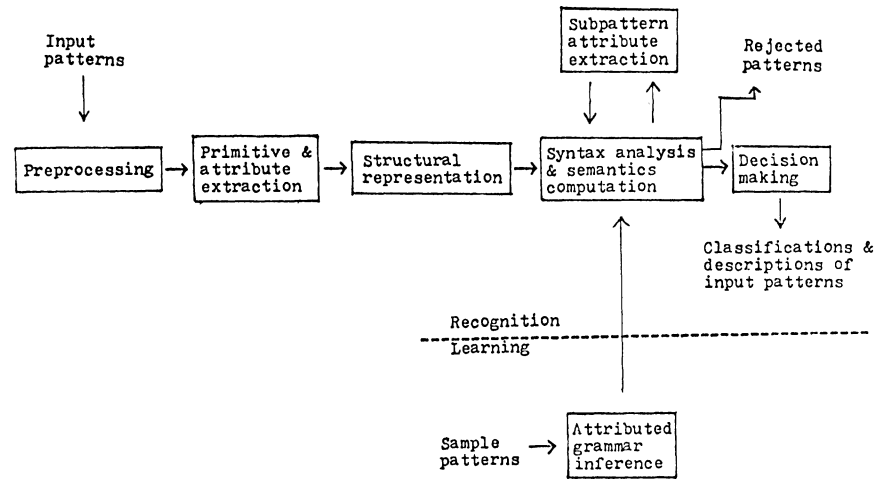


Fig. 4. Pattern analysis system using attributed grammars.

the computation of semantics using the semantic rules, some nonterminal (subpattern) attributes may not be obtainable by a mapping or computation from lower level terminal attributes, then it is necessary to go back to the input pattern, to find out the subpattern corresponding to the nonterminal, and to perform the necessary *subpattern attribute extractions* as specified in the semantic rule. A note to be emphasized here is that subpattern attributes cannot be extracted before syntax analysis and semantics computation because, without the guidance of syntax analysis, the system would not know which terminals (primitives) should be grouped into a nonterminal (subpattern). This is indeed an advantage of using attributed grammars because subpattern attribute extraction now becomes more effective with the guidance of syntax analysis. Such an advantage is not obtainable by using the statistical approach alone. After this stage, we are left with a parse of the pattern representation if the input pattern is syntactically correct, together with its total attribute vector. A syntactically incorrect representation corresponding to a structurally erroneous pattern will be rejected by the parser. The hierarchical syntactic and semantic description of the input pattern is also available now except its class assignment.

The final stage is the decisionmaking performed on the total attribute vector to classify the input pattern. The various decision criteria in statistical pattern recognition are all applicable. Actually, by combining syntax analysis, semantics computation, and decisionmaking, we have formulated a quite general and powerful scheme for pattern analysis which can be viewed as a combination of syntactic and statistical approaches to pattern recognition.

### III. STATISTICAL CONSIDERATIONS FOR USING ATTRIBUTED GRAMMARS

To use attributed grammars for statistical classification, at least the following three kinds of statistical information should be considered.

1) The occurrence probability of every pattern class: This *a priori* information usually is assigned intuitively or

collected through long-term observations of pattern occurrences.

2) The occurrence probability of a specific pattern structure within its pattern class: For example, in the example of chromosome recognition, three different string representations can be generated for the class of acrocentric chromosomes, then we would want to know the occurrence probability of each string representation which represents a distinct pattern structure. Such probabilities can be obtained and derived by using stochastic grammars [7], [14] to be discussed in Section III-A.

3) The occurrence probability distribution or density functions of the total attribute vectors of each pattern: Such information is needed for the classification in terms of semantic attributes.

The above three kinds of statistical information, though identified separately, must be combined together for a complete statistical classification of an input pattern. In the following sections, deriving an optimum classification criterion is attempted according to the Bayes decision rule, which utilizes all these three kinds of statistical information. Computations of these information according to such a criterion are discussed. Various suboptimal cases are derived.

#### A. Stochastic Grammars for Specifying Pattern Structural Occurrence Probabilities

Stochastic grammars were proposed to model the structure of noisy patterns and also to specify the occurrence probability for each pattern accepted by the grammars [7]. To be more specific, and to fit into our discussion of attributed grammars, we define stochastic attributed context-free string grammars in the following.

*Definition 2:* A stochastic attributed context-free string grammar is a 4-tuple  $G = (V_N, V_T, P, S)$  where all notations are the same as those for a nonstochastic one (Definition 1) except that each syntactic rule in each production of  $P$  is of the following form:

$$X_0 \xrightarrow{P} X_1 X_2 \cdots X_m$$

where  $X_0 \in V_N$ ,  $X_j \in (V_N \cup V_T)$  for  $1 \leq j \leq m$ , and  $p$  is the probability associated with this syntactic rule. Furthermore,  $0 < p \leq 1$ , and the summation of all the probabilities associated the syntactic rules with  $X_0$  at the left-hand side must be equal to one. The semantic rule of each production is the same as in Definition 1.

When a stochastic attributed grammar is used to generate a pattern representation, a pattern occurrence probability can also be computed simultaneously which is the product of all the probabilities associated with the syntactic rules used in deriving the pattern representation. This occurrence probability is used to specify the within-class occurrence possibility of the pattern structure (structure of the pattern representation), and this can be made possible by carefully adjusting probabilities associated with the syntactic rules [14].

Since each attributed grammar can generate several pattern classes (e.g.,  $G_S$  in Example 1 covers three classes of chromosomes), several sets of probability assignments for the syntactic rules are required, each set for a corresponding class. After the string representation  $z$  of a given unknown pattern  $\omega$  is syntactically accepted, several probabilities  $P(z|C_1), P(z|C_2), \dots, P(z|C_N)$  are computed ( $N$  = total number of classes), each  $P(z|C_i)$  being the within-class probability of  $z$  in class  $C_i$ . We will call  $P(z|C_i)$  the *structural occurrence probability* of  $z$  within class  $C_i$ , in contrast with the *attribute occurrence probability or density* to be defined later for the TAV of a given pattern. Using stochastic attributed grammars in this sense, we assume implicitly that *each pattern class syntactically contains all the strings generated by the syntactic rules of the attributed grammar*. For example, in chromosome classification, the class of acrocentric chromosomes includes all three strings  $z_1, z_2, z_3$ , generated by  $G_S$  (see the last paragraph of Example 1). However, this seems not to be the case for the class of median or submedian chromosomes; either class contains only the string  $z_1$ , but not  $z_2$  and  $z_3$ . As far as a classification is concerned, though, we can adjust the probability assignments of the syntactic rules so that unwanted strings in certain classes (such as  $z_2, z_3$  in the median or submedian class) will have low or zero occurrence probabilities computed and thus not be classified as from those classes [12].

### B. An Optimum Decision Rule for Pattern Classification Using Attributed Grammars

After an input pattern  $\omega$  is analyzed syntactically and accepted by an attributed grammar, a total attribute vector  $X$  is also obtained as the result of simultaneous semantics computation. This vector  $X$  is supposed to include all attributes, either from primitives or from sub-patterns (or even from the whole pattern), which are useful for pattern classification. If every pattern is free of noise or distortion, then  $X$  will be invariant; every component value  $x_i$  in  $X = (x_1, x_2, \dots, x_n)$  will be fixed to one possible value, and the classification of  $\omega$  can be easily accomplished by matching  $X$  with the noise-free TAV's of

the prototype or reference patterns in each pattern class. In real world situations, however, noise and distortion usually are not avoidable, and  $X$  is subject to numerical variations. Probability distribution or density functions thus can be introduced on  $X$  and a statistical classification becomes possible. This not only solves partially the problem of handling noisy or distorted patterns, which is a weakness of conventional syntactic methods, but also increases classification accuracy because optimum decision criteria, such as the Bayes rule, are applicable here, as can be seen in the following discussions.

Given an unambiguous attributed grammar  $G$ , let  $N$  be the total number of pattern classes covered by  $G$ , and  $n$  be the total number of distinct strings (corresponding to pattern structures) which can be generated by  $G$ . Here  $n$  may be infinite if there exists recursive syntactic rules in the productions of  $G$ . According to the discussions in Section III-A, it is considered that each pattern class  $C_i$  ( $i = 1, 2, \dots, N$ ) syntactically contains all the  $n$  strings  $z_1, z_2, \dots, z_n$  generated by  $G$ . Now suppose that there is no noise or distortion and we want to generate syntactic and semantic descriptions of *pure patterns*. This can be done by using, for each class  $C_i$ , a distinct set of numerical values for the terminal and nonterminal attributes used in the semantic rules and then computing the total attribute vector  $X_{ij}$  for each  $z_j$  (subscripts  $i, j$  in  $X_{ij}$  specify that this  $X_{ij}$  is computed for  $z_j$  of class  $C_i$ ). If we consider the 2-tuple  $\omega_{ij} = (z_j, X_{ij})$  as the description of a noise-free pattern in class  $C_i$ , then each  $C_i$  consists of exactly  $n$  noise-free patterns  $C_i = \{\omega_{i1}, \omega_{i2}, \dots, \omega_{in}\}$ ; but since in practice, attributes of input patterns are subject to numerical variations due to noise and distortion, each  $\omega_{ij}$  semantically can be deformed into a finite or infinite set of noisy versions. All these versions have an identical symbolic string representation  $z_j$ , but are different in their attribute values, or after semantics computation, in their TAV values. Thus for  $\omega_{ij} = (z_j, X_{ij})$  in  $C_i$ , let the set of its noisy versions be denoted as  $D(\omega_{ij}) = \{\omega_{ijk} | \omega_{ijk} = (z_j, X_{ijk}), k = 1, 2, \dots, m_{ij}\}$  where  $m_{ij}$  may be finite or infinite, then  $C_i$  can be regarded as  $C_i = D(\omega_{i1}) \cup D(\omega_{i2}) \cup \dots \cup D(\omega_{in})$ . We assume that  $\omega_{ij} \in D(\omega_{ij})$ , i.e.,  $\omega_{ij}$  is considered as a possible version in  $D(\omega_{ij})$ . Since each  $\omega_{ijk} \in D(\omega_{ij})$  may occur with a different probability, we can introduce a conditional probability distribution or density function  $p(\cdot | \omega_{ij}, C_i)$  on  $D(\omega_{ij})$  for class  $C_i$  such that

$$p(\omega_{ijk} | \omega_{ij}, C_i) = p(X_{ijk} | z_j, C_i) \quad (1)$$

is the occurrence probability or density for  $\omega_{ijk} = (z_j, X_{ijk}) \in D(\omega_{ij})$ , which will be called the *attribute occurrence probability or density* of  $\omega_{ijk}$  from  $\omega_{ij}$ , in contrast with the *structural occurrence probability* of  $z_j$  within class  $C_i$ ,  $P(z_j | C_i)$ .

Now, given an unknown pattern  $\omega$ , if its string representation is accepted syntactically by the grammar as  $z_j$ , and if the computed TAV for  $\omega$  is  $X$ , then  $\omega = (z_j, X)$  can be regarded as a noisy version of  $\omega_{ij}$ , included in  $D(\omega_{ij})$ . So, the attribute occurrence probability or density of  $\omega$

from  $\omega_{ij}$  with respect to class  $C_i$  is

$$p(\omega|\omega_{ij}, C_i) = p(X|z_j, C_i) \quad (2)$$

and the composite occurrence probability or density that  $\omega \in C_i$  is

$$\begin{aligned} p(\omega|C_i) &= p(\omega|\omega_{ij}, C_i)P(\omega_{ij}|C_i) \\ &= p(X|z_j, C_i)P(z_j|C_i), \end{aligned} \quad (3)$$

where  $P(\omega_{ij}|C_i)$  is used symbolically to specify  $P(z_j|C_i)$ .

Next, we compute the *a posteriori* probability or density,  $p(C_i|\omega)$ , that  $\omega$  can be recognized as a deformed pattern from pattern class  $C_i = \cup_{j=1}^n D(\omega_{ij})$ . Since the attributed grammar  $G$  is assumed to be unambiguous, each string representation accepted syntactically by  $G$  has only one derivation [13]. Then, if the string representation  $z$  of  $\omega$  is accepted as  $z_j$ , we get

$$\begin{aligned} p(C_i|\omega) &= p(\omega|C_i)P(C_i)/p(\omega) \\ &= p(\omega|\omega_{ij}, C_i)P(\omega_{ij}|C_i)P(C_i)/p(\omega) \\ &= p(X|z_j, C_i)P(z_j|C_i)P(C_i)/p(\omega) \end{aligned} \quad (4)$$

where  $\omega_{ij} = (z_j, X_{ij})$ ,  $P(C_i)$  is the *a priori* probability of class  $C_i$ , and  $p(\omega) = \sum_{i=1}^N p(\omega|C_i)P(C_i)$ .

When the attributed grammar  $G$  is ambiguous, the above discussion is no longer valid and should be modified as follows. First, a string accepted by  $G$  syntactically now may have several or an infinite number of distinct derivations. For the system to be implementable, we assume here that  $G$  is not *infinitely ambiguous* [28] so that each string accepted by  $G$  has only a finite number of distinct derivations. This assumption is general enough for most applications, and is more general than the unambiguity assumption made in several studies on syntactic pattern recognition [7], [14], [25]. Next, to generate the set of pure patterns, we consider each distinct derivation of a given string  $z_j$  as the structural representation of a distinct pure pattern because different derivations of  $z_j$  which use different sets of syntactic rules will result in different TAV values being computed. Therefore, if we denote the  $k$ th derivation of  $z_j$  as  $z_j^k$ , and its corresponding TAV value with respect to class  $C_i$  as  $X_{ij}^k$ , for  $k = 1, 2, \dots, l_j$ , where  $l_j$  is the total number of distinct derivations of  $z_j$ , then each  $\omega_{ij}^k = (z_j^k, X_{ij}^k)$  can be regarded as a pure pattern in class  $C_i$ . Again, denoting the set of all noisy or deformed versions of  $\omega_{ij}^k$  as  $D(\omega_{ij}^k)$ , we get  $C_i = \cup_{j=1}^n \cup_{k=1}^{l_j} D(\omega_{ij}^k)$  where  $n$  is the total number of distinct strings generated by  $G$ .

Now, given an unknown pattern  $\omega$  with string representation  $z$ , if  $z$  is accepted by  $G$  as  $z_j$ , then the  $k$ th of the  $l_j$  derivations of  $z = z_j$ ,  $z_j^k$ , together with the corresponding TAV value,  $X^k$ , form a 2-tuple  $(z_j^k, X^k)$  which can be regarded as a noisy version of  $\omega_{ij}^k$ , i.e.,  $(z_j^k, X^k) \in D(\omega_{ij}^k)$ . Furthermore, because of the different combinations of syntactic rules used in the derivations, each different derivation  $z_j^k$  of  $z = z_j$  will also result in a different structural occurrence probability value, which we denote as  $P(z_j^k|C_i)$ . Let  $p(X^k|z_j^k, C_i)$  be the attribute occurrence probability or density of  $(z_j^k, X^k)$  from  $\omega_{ij}^k$ , defined for

the  $k$ th derivation of  $z_j$ , then the composite probability or density that  $\omega \in C_i$  is

$$\begin{aligned} p(\omega|C_i) &= \sum_{k=1}^{l_j} [p(\omega|\omega_{ij}^k, C_i)P(\omega_{ij}^k|C_i)] \\ &= \sum_{k=1}^{l_j} [p(X^k|z_j^k, C_i)P(z_j^k|C_i)] \end{aligned} \quad (5)$$

where the summation in (5) is justified on the basis that  $\omega$  can be regarded as coming from any of the  $l_j$  pure patterns  $\omega_{ij}^1, \omega_{ij}^2, \dots, \omega_{ij}^{l_j}$ , as discussed previously. Note that the identical superscript  $k$  in the right side of (5) specifies that both  $p(X^k|z_j^k, C_i)$  and  $P(z_j^k|C_i)$  should be evaluated with respect to the same ( $k$ th) derivation of  $z_j$ .

Finally, the Bayes decision rule to classify the unknown pattern  $\omega$  is as follows:

$$\text{assign } \omega \text{ to } C_M \text{ if } p(C_M|\omega) = \max_{i=1,2,\dots,N} p(C_i|\omega).$$

Or, after applying Bayes Theorem and (5),

$$\begin{aligned} \text{assign } \omega \text{ to } C_M \text{ if } & \sum_{k=1}^{l_j} [p(X^k|z_j^k, C_M)P(z_j^k|C_M)]P(C_M) \\ & = \max_{i=1,2,\dots,N} \sum_{k=1}^{l_j} [p(X^k|z_j^k, C_i)P(z_j^k|C_i)]P(C_i) \end{aligned} \quad (6)$$

where  $p(\omega)$ , common to both sides, has been dropped. The above Bayes classification rule is for one grammar only. Extension to cases with more than one grammar is trivial. That is,

$$\text{assign } \omega \text{ to } C_M \text{ if } p(C_M|\omega) = \max_{G_j} \max_{i=1,2,\dots,N_{G_j}} p(C_i|\omega), \quad (7)$$

where  $N_{G_j}$  is the total number of pattern classes covered by grammar  $G_j$ .

### C. Computation of Occurrence Probabilities or Densities, and String Structure Identification Problems

To compute the structural occurrence probability  $P(z_j^k|C_i)$  in the Bayes classification rule (6), recall that we are using stochastic attributed grammars in which each syntactic rule has been associated with a probability. Let the sequence of syntactic rules used in the  $k$ th derivation of  $z_j$  be  $\gamma_1, \gamma_2, \dots, \gamma_R$ , and the probability associated with the rule  $\gamma_h$  be  $p_h$ ,  $h = 1, 2, \dots, R$ , then

$$P(z_j^k|C_i) = p_1 p_2 \dots p_R. \quad (8)$$

Equation (8) can be computed during parsing by multiplying each  $p_h$  of a newly used syntactic rule  $\gamma_h$  to the product of those of previously used rules. So  $P(z_j^k|C_i)$  can be obtained for each  $C_i$  right at the end of parsing, *but the case for computing the attribute occurrence probability or density  $p(X^k|z_j^k, C_i)$  on the contrary is not so simple*. First, this distribution depends on the syntactic structure of the recognized string representation  $z_j$ . Given two input patterns from class  $C_i$ , assume that they have the same total

attributed vector  $X^k$  computed. If they are accepted syntactically to have different string structures  $z_{j_1}$  and  $z_{j_2}$ , the probabilities or densities used for them should be  $p(X^k|z_{j_1}^k, C_i)$  and  $p(X^k|z_{j_2}^k, C_i)$ , respectively, which may be different; but since in general the string structure ( $z_{j_1}$  or  $z_{j_2}$ ) can be determined only *after* the whole string representation of the input pattern is parsed, attribute occurrence probabilities or densities can *not* be computed during the parsing procedure. Computation of such values must be delayed until parsing is completed *and the structure of the input string representation is identified*. We must now face this *string structure identification problem*: how can the system know, at the end of parsing, a syntactically accepted string representation is  $z_{j_1}$  or  $z_{j_2}$  so that a correct conditional probability or density function can be chosen to compute the attribute occurrence probability or density?

This problem is not unique to our case of using attributed grammars here. It actually exists whenever probability distributions or density functions for primitives (terminals) or subpatterns (nonterminals) are used for pattern classification or linguistic analysis [10], [15]–[18]. So far, it has always been assumed that the occurrence probability of a certain primitive (terminal) is invariant with respect to different patterns (linguistic sentences), or more specifically, that two identical primitives (terminals) used in two different patterns (linguistic sentences) are regarded to have the same occurrence probability. Using our notations here, this means that  $p(X^k|z_{j_1}^k, C_i) = p(X^k|z_{j_2}^k, C_i) = p(X^k/C_i)$ . Such an assumption, though simplifying statistical discussion, theoretically is not general enough for applications using attributed grammars. To solve this problem, it seems that we can separate an attributed grammar into several ones so that each grammar generates a set of strings  $z_1, z_2, \dots, z_n$  which make the following equality to be true:

$$p(X^k|z_1^k, C_i) = p(X^k|z_2^k, C_i) = \dots = p(X^k|z_n^k, C_i).$$

A trivial case is when  $n=1$ . This solution is impractical because it destroys the essence of grammar usage—with one grammar to efficiently cover as many structurally similar patterns as possible. Observing the simple fact that if two strings are different, they must be respectively derived from at least two different syntactic rules, we propose the following solution. Add to the attribute grammar a set of rules, called *identification rules*, which are similar to the semantic rules. Each of these rules assigns or computes some specific discrete numerical value to a certain artificially created *identification variable* (like an attribute) when a corresponding syntactic rule is used in deriving a string representation during parsing. The identification variable values are computed in the same manner as the semantics computation. At the end of parsing, the discrete values of the identification variables  $i_S$  of the starting symbol  $S$  are used, in a table look-up manner, to identify the syntactic structure of the input string representation. Of course, the numbers of identification rules and identification variables should be kept as small as

possible to reduce extra processing time. We give an illustrative example, though not complicated, in the following which also explains the usage of the Bayes classification rule (6) proposed in Section III-B.

*Example 2: Statistical Classification of Chromosomes (Continuation of Example 1):* To use the Bayes classification rule (6) to classify an unknown input chromosome, we first assume the following *a priori* class probabilities:

$$P(C_1) = 0.4 \text{ for } C_1 = \text{median class}$$

$$P(C_2) = 0.3 \text{ for } C_2 = \text{submedian class}$$

$$P(C_3) = 0.3 \text{ for } C_3 = \text{acrocentric class.}$$

Next, the syntactic rules of the chromosome grammar  $G_S$  are made stochastic as in the following. On the other hand, to solve the string structure identification problem which is simple here, we also add three identification rules to the productions of  $G_S$ :

syntactic rules	semantic rules	identification rules
1) $S \xrightarrow{p_{i1}} Q_1^1 Q_1^2$	$l_{S1} \leftarrow l_{Q_1^1}, l_{S2} \leftarrow l_{Q_1^2}$	$i_S \leftarrow 1$
2) $S \xrightarrow{p_{i2}} Q_1 Q_2$	$l_{S1} \leftarrow l_{Q_1}, l_{S2} \leftarrow l_{Q_2}$	$i_S \leftarrow 2$
3) $S \xrightarrow{p_{i3}} Q_2 Q_1$	$l_{S1} \leftarrow l_{Q_2}, l_{S2} \leftarrow l_{Q_1}$	$i_S \leftarrow 3$
4) $Q_1 \xrightarrow{1.0} DR_1$	$l_{Q_1} \leftarrow l_{R_1}$	
5) $Q_2 \xrightarrow{1.0} DR_2$	$l_{Q_2} \leftarrow l_{R_2}$	
6) $R_1 \xrightarrow{1.0} M_1^1 CM_1^2$	$l_{R_1} \leftarrow (l_{M_1^1} + l_{M_1^2})/2$	
7) $R_2 \xrightarrow{1.0} M_2^1 CM_2^2$	$l_{R_2} \leftarrow 0$	
8) $M_1 \xrightarrow{1.0} B^1 AB^2$	$l_{M_1} \leftarrow (l_{B^1} + l_{B^2})/2$	
9) $M_2 \xrightarrow{1.0} A$		
10) $A \xrightarrow{1.0} a$		
11) $B \xrightarrow{1.0} b$	$l_B \leftarrow l_b$	
12) $C \xrightarrow{1.0} c$		
13) $D \xrightarrow{1.0} d$		

The subscript  $i$  in each  $p_{ij}$  ( $i=1,2,3$ ) is provided to indicate that  $p_{ij}$  are different for different pattern classes  $C_j$ . Recall that the above syntactic rules can generate three different string representations (see the last paragraph of Example 1):

$$z_1 = dbabcbabdbabcbab \text{ (chromosome of any class)}$$

$$z_2 = dbabcbabdaca \text{ (acrocentric chromosome)}$$

$$z_3 = dacadbabcbab \text{ (acrocentric chromosome).}$$

Obviously, after parsing, we have the following corresponding relations between input string structures ( $z$ ) and identification variable values ( $i_S$ ):

$$i_S = 1 \leftrightarrow z = z_1$$

$$i_S = 2 \leftrightarrow z = z_2$$

$$i_S = 3 \leftrightarrow z = z_3.$$

Now assuming the following probability assignment for



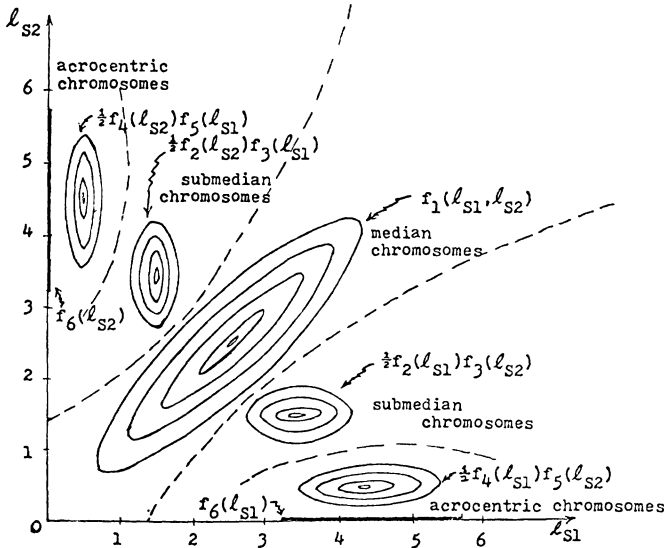


Fig. 5. Probability density functions for arm pair lengths of various chromosomes (dotted lines are approximate class decision boundaries).

the syntactic rules:

$$\begin{aligned} p_{11} &= 1.0, p_{12} = 0.0, p_{13} = 0.0 \\ p_{21} &= 1.0, p_{22} = 0.0, p_{23} = 0.0 \\ p_{31} &= 0.2, p_{32} = 0.4, p_{33} = 0.4 \end{aligned}$$

and noting that  $P(z_1|C_i) = p_{i1}$ ,  $P(z_2|C_i) = p_{i2}$ ,  $P(z_3|C_i) = p_{i3}$  for  $i=1,2,3$ , we get the following structural occurrence probabilities:

$$\begin{aligned} P(z_1|C_1) &= 1.0, P(z_1|C_2) = 1.0, P(z_1|C_3) = 0.2 \\ P(z_2|C_1) &= 0.0, P(z_2|C_2) = 0.0, P(z_2|C_3) = 0.4 \\ P(z_3|C_1) &= 0.0, P(z_3|C_2) = 0.0, P(z_3|C_3) = 0.4. \end{aligned}$$

Based on the probabilities computed, it is easy to see that  $z_2, z_3$  which are acrocentric chromosomes are accepted, as desired, by class  $C_3$  only with probability 0.4, but  $z_1$  is accepted by all three classes with probabilities 1.0, 1.0, and 0.2, respectively. This is due to the fact that chromosomes from any class may happen to have the same syntactic structure as  $z_1$ . A discrimination between them will then rely mostly on the semantic attributes. Recalling that the chromosome grammars  $G_S$  (in Example 1) synthesizes the TAV  $X = (l_{S1}, l_{S2})$  for an input pattern  $\omega$  with  $l_{S1}, l_{S2}$  being the lengths of left-hand and right-hand arm pairs, we assume the following attribute occurrence density functions:

$$\begin{aligned} p(X|z_1, C_1) &= f_1(l_{S1}, l_{S2}), \quad \text{for } C_1 \\ p(X|z_1, C_2) &= \frac{1}{2}f_2(l_{S1})f_3(l_{S2}) + \frac{1}{2}f_2(l_{S2})f_3(l_{S1}), \quad \text{for } C_2 \\ p(X|z_1, C_3) &= \frac{1}{2}f_4(l_{S1})f_5(l_{S2}) + \frac{1}{2}f_4(l_{S2})f_5(l_{S1}), \quad \text{for } C_3 \\ p(X|z_2, C_3) &= f_6(l_{S2}), \quad \text{for } C_3 \\ p(X|z_3, C_3) &= f_6(l_{S1}), \quad \text{for } C_3 \end{aligned}$$

where  $f_1, f_2, \dots, f_6$  are all Gaussian density functions:

- $f_1$  is bivariate with means 2.5, 2.5, variances  $0.5^2, 0.5^2$ , and correlation coefficient 0.6;

- $f_2, \dots, f_6$  are all univariate with means 3.5, 1.5, 4.5, 0.5, 4.5, respectively, and with variances  $0.30^2, 0.20^2, 0.35^2, 0.15^2, 0.40^2$ , respectively.

The  $p(X|z_j, C_i)$  and the approximate decision boundaries between different chromosomes are sketched in Fig. 5.  $p(X|z_2, C_1), p(X|z_2, C_2), p(X|z_3, C_1), p(X|z_3, C_2)$  are not specified because  $P(z_2|C_1), P(z_2|C_2), P(z_3|C_1), P(z_3|C_2)$  are all zeros.

Now suppose that we want to classify an unknown input chromosome  $\omega$  with string representation  $z = dbabcbabdbabcbab$  which is produced after preprocessing and primitive extraction. Also assume that the synthesized TAV  $X = (5.5, 0.4)$ . The synthesized identification variable  $i_S$  is 1, which means that the syntactic structure of  $z$  is  $z_1$ . Also, after parsing, the structural occurrence probability obtained is  $P(z|C_i) = p_{i1}$ .<sup>3</sup> Before classification, we calculate the following attribute occurrence density values:

$$\begin{aligned} p(X|z_1, C_1) &= f_1(5.5, 0.4) = 1.36 \times 10^{-23} \\ p(X|z_1, C_2) &= \frac{1}{2}f_2(5.5)f_3(0.4) + \frac{1}{2}f_2(0.4)f_3(5.5) \\ &= 1.60 \times 10^{-16} \\ p(X|z_1, C_3) &= \frac{1}{2}f_4(5.5)f_5(0.4) + \frac{1}{2}f_4(0.4)f_5(5.5) \\ &= 2.05 \times 10^{-2}. \end{aligned}$$

Now to apply the Bayes classification rule (6), we have

$$\begin{aligned} p(X|z_1, C_1)P(z_1|C_1)P(C_1) &= 1.36 \times 10^{-23} \times 1.0 \times 0.4 = 5.44 \times 10^{-24} \\ p(X|z_1, C_2)P(z_1|C_2)P(C_2) &= 1.60 \times 10^{-16} \times 1.0 \times 0.3 = 4.80 \times 10^{-17} \\ p(X|z_1, C_3)P(z_1|C_3)P(C_3) &= 2.05 \times 10^{-2} \times 0.2 \times 0.3 = 1.23 \times 10^{-4} \end{aligned}$$

Therefore, the decision is that the input pattern  $\omega$  is an acrocentric chromosome. As can be seen, the contribution to this statistical decision mostly comes from the attribute occurrence density values, though  $\omega$  has a pattern structure  $z_1$  which normally is a median or submedian chromosome structure.

#### D. Least-Square-Error (LSE) Classification

In this section, we derive various least-square-error decision criteria for pattern classification using attributed grammars, which are practical for applications when inference of probability distribution or density functions is difficult or impossible, although they are just special cases of the optimum Bayes classification rule (6).

First, recall the Bayes classification rule (6):

$$\begin{aligned} \text{assign } \omega \text{ to } C_M \text{ if } & \sum_{k=1}^{I_j} [p(X^k|z_j^k, C_M)P(z_j^k|C_M)]P(C_M) \\ &= \max_{i=1,2,\dots,N} \sum_{k=1}^{I_j} [p(X^k|z_j^k, C_i)P(z_j^k|C_i)]P(C_i). \quad (6) \end{aligned}$$

Now assume the following conditions.

- 1) All pattern classes have an equal *a priori* class probability  $P(C_i)$ .
- 2) The grammar is unambiguous; only one parse or derivation exists for each input pattern representation  $z$ . So  $l_j=1$ , and the superscript  $k$  can be removed.
- 3) The structural occurrence probabilities  $P(z_j|C_i)$  are equal for all possible  $z_j$  and all  $i=1,2,\dots,N$ . This implies that we can use nonstochastic syntactic rules.
- 4) The string structure identification problem is ignored. That is, we assume  $p(X|z_j, C_i)=p(X|C_i)$  for all  $j$ .
- 5) Each component  $x_h$  in  $X=(x_1, x_2, \dots, x_m)$  is independently and normally distributed with mean  $m_h^i$  and variance  $(\sigma_h^i)^2$  for class  $C_i$ , or

$$p(X|C_i) = \prod_{h=1}^m \frac{1}{\sqrt{2\pi} \sigma_h^i} \exp \left[ -\frac{1}{2} \cdot \frac{(x_h - m_h^i)^2}{(\sigma_h^i)^2} \right].$$

Under these conditions, the Bayes rule (6) becomes

$$\text{assign } \omega \text{ to } C_M \text{ if } p(X|C_M) = \max_{i=1,2,\dots,N} p(X|C_i).$$

After taking negative logarithms and removing constant terms, we get

assign  $\omega$  to  $C_M$  if

$$\sum_{h=1}^m \left[ (x_h - m_h^M)^2 / (\sigma_h^M)^2 - 2 \ln \sigma_h^M \right] \\ = \min_{i=1,2,\dots,N} \sum_{h=1}^m \left[ (x_h - m_h^i)^2 / (\sigma_h^i)^2 - 2 \ln \sigma_h^i \right] \quad (9)$$

which we call the *normalized least-square-error (NLSE) classification rule*. Ignoring  $\sigma_h^i$  or assuming all  $\sigma_h^i = 1$  in (9), we get

assign  $\omega$  to  $C_m$  if

$$\sum_{h=1}^m (x_h - m_h^M)^2 = \min_{i=1,2,\dots,N} \sum_{h=1}^m (x_h - m_h^i)^2, \quad (10)$$

which we call the *unnormalized least-square-error (ULSE) classification rule*. If we remove the logarithm terms in (9) and replace the variance  $(\sigma_h^i)^2$  by a weight  $(1/w_h^i)$ , we then get

assign  $\omega$  to  $C_M$  if

$$\sum_{h=1}^m w_h^M \cdot (x_h - m_h^M)^2 = \min_{i=1,2,\dots,N} \sum_{h=1}^m w_h^i \cdot (x_h - m_h^i)^2, \quad (11)$$

which we will call the *weighted least-square-error (WLSE) classification rule*.

When the inference of variances  $(\sigma_h^i)^2$  or weights  $w_h^i$  is difficult, the simplest ULSE criterion can be used. If the values of  $(\sigma_h^i)^2$  are available, the NLSE criterion gives better results. Otherwise, we can also assign the weights  $w_h^i$  subjectively, and use the WLSE criterion. The ULSE and NLSE criteria have been used by Tsai and Fu [19] in syntactic texture discrimination.

TABLE I  
CLASSES OF I WRENCHES (EXTRACTED SIZES AND LENGTHS  
ARE IN PIXEL NUMBERS)

Class Number	I Wrench Brand	Real Sizes		Extracted Sizes ( $d_1$ and $d_2$ )		Extracted Length ( $l$ )
1	Craftsman	1/4	5/16	3.61	5.09	81.65
2	"	3/8	7/16	6.35	7.81	90.44
3	"	1/2	9/16	9.28	10.75	104.11
4	"	5/8	11/32	12.88	16.16	130.48
5	Gedore	5/16	11/32	4.43	5.21	63.99
6	"	1/2	9/16	9.08	10.40	94.52
7	"	19/32	11/16	11.92	14.24	104.37
8	"	5/8	3/4	12.62	15.47	120.22

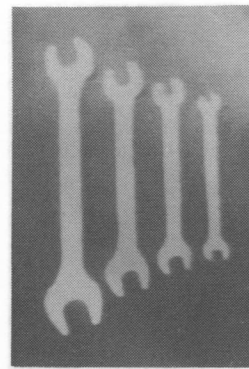


Fig. 6. Set of I wrenches.

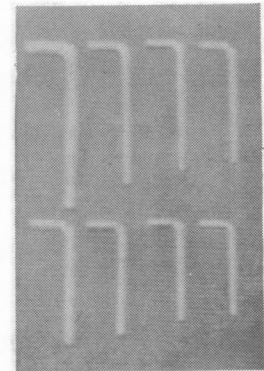


Fig. 7. Set of L wrenches.

#### IV. AN APPLICATION EXAMPLE—RECOGNITION OF MACHINE TOOLS AND EXPERIMENTAL RESULTS

Experiments have been conducted on the recognition of machine tools which are similar in shape but different in attributes, such as sizes, lengths, widths, etc. Attributed grammars are most suitable for, though not limited to, use in description and classification of such objects. Two kinds of tools used in the experiments are shown in Figs. 6 and 7, which for simplicity are called I wrenches and L wrenches, respectively, according to their shapes. Each kind of wrench further consists of several distinct wrenches with different sizes and lengths. Each distinct wrench with a fixed size and length is regarded as coming from a pattern class. There are eight classes of I wrenches and also eight classes of L wrenches (see Tables I and II). Since all classes of I wrenches are of the same shape structure, one grammar is enough for structural description of them; discrimination among the different classes will depend on semantic information—the size and length attributes—and the extraction of semantic information is specified by the semantic rules. This is also true for classes of L wrenches. The syntactic rules inferred are nonstochastic. The wrench classification is performed according to the unnormalized least-square-error criterion (Section III-D).

Images of wrenches are obtained from a TV camera. The wrenches are placed on a dark background so that high contrast images can be obtained. Each image is digitized with the resolution of  $200 \times 128$  in pixels, and

TABLE II  
CLASSES OF L WRENCHES (EXTRACTED ARM LENGTHS  
ARE IN PIXEL NUMBERS)

Class Number	I Wrench Categories	Size Order Number	Extracted Arm Length $l_1$	Extracted Arm Length $l_2$
9	with longer arms	1	24.56	88.90
10	"	2	19.08	76.36
11	"	3	18.03	69.12
12	"	4	16.65	62.99
13	with shorter arms	1	20.71	61.46
14	"	2	18.79	56.14
15	"	3	16.84	50.87
16	"	4	15.55	47.23

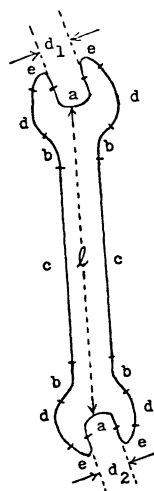


Fig. 8. I wrench and its boundary primitives.

gray levels from 0 to 63. To concentrate on the recognition work, only simple thresholding and boundary following techniques are applied for shape contour extraction [20]. The result is a chain of consecutive boundary pixel coordinates, which is then segmented into primitives. The method used for such boundary segmentation is similar to that used in the example of chromosome classification (Section II-A), i.e., grouping consecutive boundary pixels with similar curvatures into segments [11], [12]. For a graphical illustration as to how a boundary is segmented into primitives, see Figs. 8 and 9. Note that syntactic symbols 'a' and 'e' correspond to a very concave and a very convex segment, respectively. The pattern structural representation  $z$  for each wrench is simply the output string of symbols  $c_1, c_2, \dots, c_m$  of a primitive extraction algorithm described in [27], or

$$z = c_1 c_2 \dots c_m,$$

where  $c_i$  represents the structure of the corresponding primitive. To facilitate grammatical inference and parsing, symbols in the string are permuted such that  $c_1$  is always an 'a' primitive (corresponding to a very concave curve segment). Thus the string representation for an I wrench is

$$z_I = aedbcbbdeaedbcbbde$$

and that for an L wrench is

$$z_L = acececec.$$

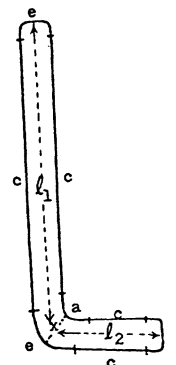


Fig. 9. L wrench and its boundary primitives.

Next, we briefly explain the procedure of extracting wrench sizes and lengths. For an I wrench, the distance ( $d_1$  or  $d_2$  in Fig. 8) between the two inner edges of either wrench head is measured as one of its two sizes. To discriminate further the brand (Craftsman or Gedore) of an I wrench, the distance between the two most concave pixels in the two 'a' primitives ( $l$  in Fig. 8) is measured as its length. For an L wrench, the lengths  $l_1, l_2$  of its two arms as shown in Fig. 9 are measured for the L wrench class discrimination. The point  $X$  in the L wrench of Fig. 9 is the middle point of the line segment joining the most concave point of the 'a' primitive and the most convex point of the 'e' primitive on the bend portion. The parsers used in the syntax analysis are deterministic finite-state automata constructed from the attributed grammars inferred for the wrenches [13], which are very efficient in parsing. They are written as Fortran programs and run on a PDP 11/45 computer.

Finally, to apply the ULSE classification criterion, we need the means of the attributes  $d_1, d_2, l$  for each of the eight classes of I wrenches and the means of  $l_1, l_2$  for each of the eight classes of L wrenches. Eight images at various orientations are taken for each class of wrenches to infer such means by averaging. The results are included in Tables I and II. 248 test images are then used for recognition. 231 images are correctly recognized. Average time for recognizing each image is 6.24 s on a PDP-11/45 computer. A recognition rate  $231/248=93.2$  percent is reached. No misclassification happens. Only syntactical rejections are observed, which indicates that a more powerful boundary smoothing technique might be needed. This also suggests the necessity of using error-correcting parsing [15]–[17] for syntax analysis.

## V. REMARKS AND CONCLUSIONS

A hybrid approach to pattern recognition using attributed grammars is proposed in this paper. In practical applications, pattern classes can often be divided into groups, each group consisting of several pattern classes which are similar in structure but different in attributes. In such cases, it is appropriate to construct an attributed grammar for each group of pattern classes, leaving the discrimination of within-group pattern classes to statistical classification on attributes. In one extreme, all pattern

classes in a pattern recognition problem may form only a single group due to structural similarity. Then class discrimination can be made to depend entirely on attribute differences without using grammars. This is the conventional statistical approach to pattern recognition. At the other extreme, there may exist no structural similarity between any two pattern classes. Then each pattern class can be described by a nonattributed grammar, and class discrimination will thus rely only on conventional grammatical analysis. As pointed out in [29], examples found in the literature [7], [12], [14], [29], [30] usually lie in between the two extremes, with a single grammar used to describe the structures of several pattern classes.

On the other hand, within each pattern class, there may exist several pattern structures represented by the strings generated by the attributed grammar. Through the use of identification rules, these pattern structures can be identified, at the end of syntax analysis, to facilitate the choice and computation of attribute occurrence probability or density values  $p(X^k|z_j^k, C_i)$ . It seems that such pattern structure identification may also be helpful in pattern classification, but this is not true with respect to within-group pattern class assignment due to the following reason. In Section III-A we have assumed that each string generated by an attributed grammar represents the structure of a pattern which belongs to each of the pattern classes within the group described by the attributed grammar. Therefore, although the string of an unknown pattern may be structurally identified using the identification rules, the unknown pattern still cannot be assigned to any pattern class within the group without further statistical classification on attributes. In short, string structures are useful in intergroup class discrimination, but they do not help in intragroup class discrimination.

Following the previous discussions, it is worth mentioning that while implementing a pattern analysis system using attributed grammars, in general, syntactic rules are inferred first to describe a group of structurally similar pattern classes, and semantic rules are then provided to extract relevant attributes for intragroup statistical classification, followed by the inference of attribute occurrence probability distribution or density functions from the extracted attributes. In other words, the inference of attributed grammars precedes the inference of statistical distributions. From the syntactic point of view, the injection of primitive and subpattern attributes into the grammatical analysis offers the following advantages over the conventional syntactic methods.

1) Flexibility in choosing primitives and subpatterns: We can easily choose those structurally apparent entities as primitives and subpatterns, leaving all nonstructural characteristics to be extracted numerically as attributes.

2) Improvement of recognition accuracy: Since attributes can be treated by introducing statistical considerations, it is not necessary to threshold, as is usually done by conventional syntactic methods, continuous data into discrete levels, which in general decreases resulting recognition accuracy.

3) Capability of recognizing noisy patterns: Random noise now can be taken care of by continuous statistics instead of just being transformed into discrete symbolic errors such as insertions, deletions, or substitutions [17].

4) Reduction of grammatical complexity: The use of attributes usually will reduce the complexity levels of the pattern grammars. In Example 1, the conventional context-free chromosome grammar is reduced to a finite-state grammar. In [9], it is found that context-sensitive shape grammars can be reduced to context-free forms or even finite-state ones with the same pattern descriptive power by introducing proper attributes. Note that a reduction of grammatical complexity usually implies improvement on recognition speed.

On the contrary, from the statistical classification point of view, we can also find several advantages in incorporating syntax analysis into statistical decisionmaking.

1) Utilization of structural information for pattern description: All such information can be incorporated into the pattern grammar by carefully inferring the syntactic production rules.

2) Effective extraction of subpattern attributes: We here pointed out that some of the difficulty encountered in statistical approaches during feature extraction stems from the lack of structural guidance. In the proposed hybrid approach, the syntactic rules and the parsing procedures offer such guidance.

3) Description and generation of patterns: This advantage of syntactic approaches is even improved when numerical attributes are added to the generated pattern descriptions.

From the above discussion, we see that the attributed grammar indeed is a good tool for combining syntactic and statistical pattern recognition. The proposed pattern analysis system using attributed grammars is also shown by experiments to be effective. Further investigations should be directed to the inference of attributed grammars, especially of the semantic rules and identification rules which seem to be problem-dependent. Higher dimensional and error-correcting attributed grammars should also be interesting subjects of study.

## REFERENCES

- [1] V. Grenander, "A unified approach to pattern analysis," in *Advances in Computers*, vol. 10. New York: Academic, 1970.
- [2] U. Kanal and B. Chandrasekaran, "On linguistic, statistical, and mixed models for pattern recognition," in *Frontiers of Pattern Recognition*, S. Watanabe, Ed. New York: Academic, 1972.
- [3] C. H. Chen, "On statistical and structural feature extraction," presented at the Joint Workshop on Pattern Recognition and Artificial Intelligence, Hyannis, MA, June 1976.
- [4] F. W. Blackwell, "Combining mathematical and structural pattern recognition," in *Proc. 2nd Int. Joint Conf. Pattern Recognition*, Copenhagen, Denmark, 1974.
- [5] D. E. Knuth, "Semantics of context-free languages," *J. Math. Syst. Theory*, vol. 2, pp. 127-146, 1968.
- [6] A. C. Shaw, "A formal picture description scheme as a basis for picture processing systems," *Inform. Contr.*, vol. 14, pp. 9-52, 1969.
- [7] K. S. Fu, *Syntactic Methods in Pattern Recognition*. New York: Academic, 1974.

- [8] G. Y. Tang and T. S. Huang, "A syntactic-semantic approach to image understanding and creation," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-1, Apr. 1979.
- [9] K. C. You and K. S. Fu, "A syntactic approach to shape recognition using attributed grammars," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-9, June 1979.
- [10] W. H. Tsai and K. S. Fu, "A pattern deformational model and Bayes error-correcting recognition system," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-9, Dec. 1979.
- [11] R. S. Ledley, L. S. Rotolo, T. J. Golab, J. D. Jacobsen, M. D. Ginsburg, and J. B. Wilson, "FIDAC: Film input to digital automatic computer and associated syntax-directed pattern recognition programming system," in *Optical and Electro-Optical Information Processing*, J. T. Tippette, et al., Eds. Cambridge, MA: MIT Press, 1965, pp. 591-614.
- [12] H. C. Lee and K. S. Fu, "A Stochastic Syntax Analysis Procedure and Its Application to Pattern Recognition," *IEEE Trans. Comput.*, vol. C-21, pp. 660-666, 1972.
- [13] A. V. Aho and J. D. Ullman, *The Theory of Parsing, Translation and Compiling*, vol. 1. Englewood Cliffs, NJ: Prentice-Hall, 1972.
- [14] H. C. Lee and K. S. Fu, "A syntactic pattern recognition system with learning capability," in *Proc. 4th Int. Symp. Computers and Information Science*, Miami Beach, FL, Dec. 1972.
- [15] L. W. Fung and K. S. Fu, "Stochastic syntactic decoding for pattern classification," *IEEE Trans. Comput.*, vol. C-24, July 1975.
- [16] R. A. Thompson, "Language correction using probabilistic grammars," *IEEE Trans. Comput.*, vol. C-25, Mar. 1976.
- [17] S. Y. Lu and K. S. Fu, "Stochastic Error-Correcting Syntax Analysis for Recognition of Noisy Patterns," *IEEE Trans. Comput.*, vol. C-26, Dec. 1977.
- [18] V. A. Kovalevsky, "Sequential optimization in pattern recognition and pattern description," in *Proc. Int. Federal Information Processing Congr.*, Amsterdam, the Netherlands, 1968.
- [19] W. H. Tsai and K. S. Fu, "Image segmentation and recognition by texture discrimination: A syntactic approach," in *Proc. 4th Int. Joint Conf. Pattern Recognition*, Kyoto, Japan, Dec. 1978.
- [20] R. O. Duda and P. E. Hart, *Pattern Classification and Scene Analysis*. New York: Wiley, 1972.
- [21] T. Pavlidis, *Structural Pattern Recognition*. New York: Springer, 1977, pp. 168.
- [22] K. S. Fu, "Syntactic pattern recognition and stochastic languages," in *Frontier of Pattern Recognition*, S. Watanabe, Ed. N.Y.: Academic, 1972.
- [23] M. L. Baird and M. D. Kelly, "A paradigm for semantic picture recognition," *Pattern Recognition*, vol. 6, 1974.
- [24] W. H. Tsai and K. S. Fu, "Error-correcting isomorphisms of attributed relational graphs for pattern analysis," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-9, Dec. 1979.
- [25] K. C. You and K. S. Fu, "Syntactic shape recognition using attributed grammars," Purdue University, Tech. Rep. TR-EE 78-38, Aug. 1978.
- [26] A. C. Shaw, "Parsing of graph-representable pictures," *J. Ass. Comput. Mach.*, vol. 17, July 1970.
- [27] W. H. Tsai and K. S. Fu, "Recognition of patterns with syntactic and semantic deformations," Purdue University, Tech. Rep. TR-EE 79-48, Dec. 1979.
- [28] M. A. Harrison, *Introduction to Formal Language Theory*. San Francisco, CA: McGraw-Hill, 1978.
- [29] A. J. Filipinski, "A least mean-squared error approach to syntactic classification," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-2, May 1980.
- [30] W. H. Tsai and K. S. Fu, "A syntactic-statistical approach to recognition of industrial objects," in *Proc. 5th Int. Conf. Pattern Recognition*, Miami Beach, FL, Dec. 1-4, 1980.