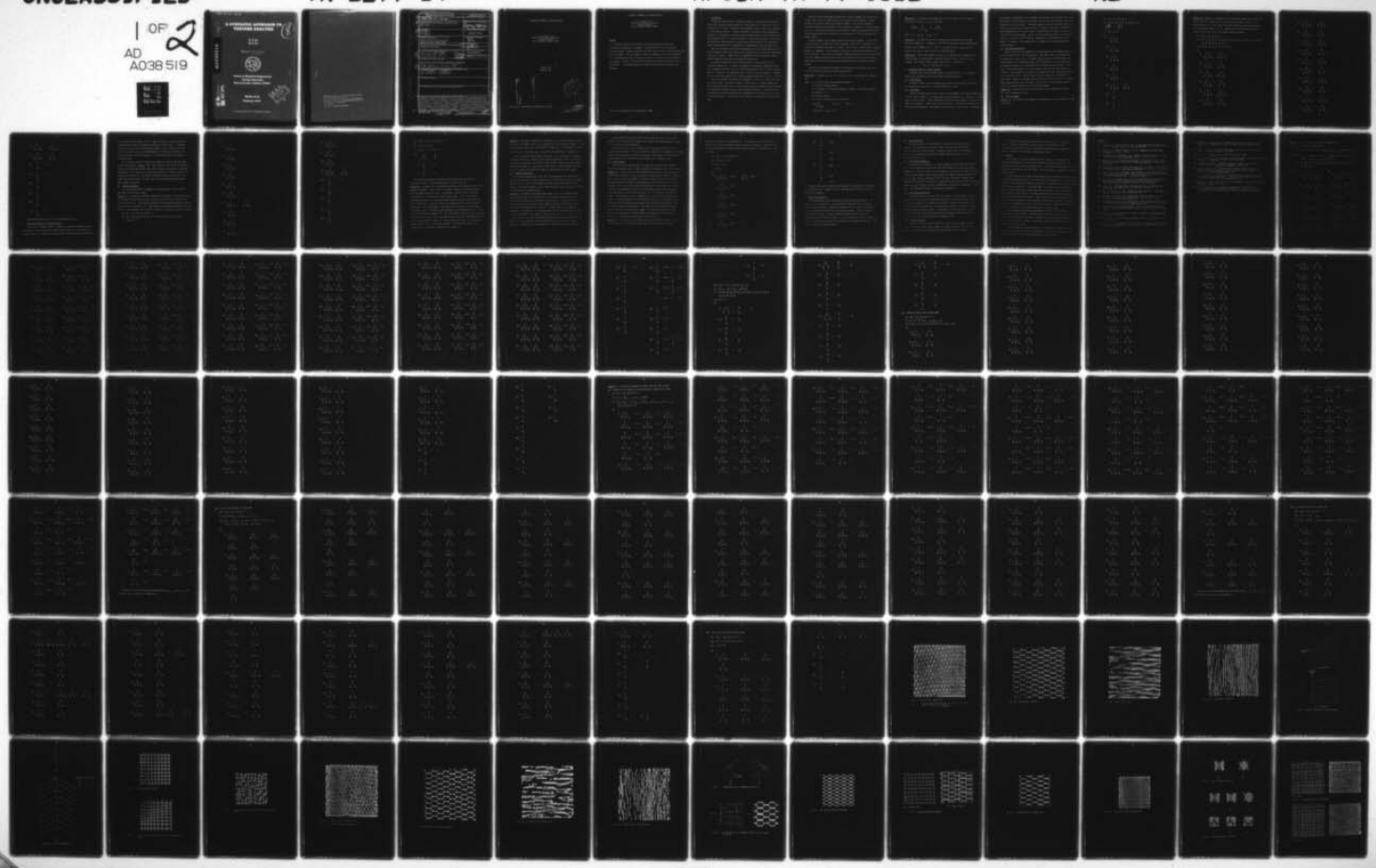
AD-A038 519 PURDUE UNIV LAFAYETTE IND SCHOOL OF ELECTRICAL ENGI--ETC F/G 12/2
A SYNTACTIC APPROACH TO TEXTURE ANALYSIS.(U)
FEB 77 S Y LU, K S FU AF-AFOSR-2661-74

UNCLASSIFIED TR-EE77-14 AFOSR-TR-77-0382 NL



A SYNTACTIC APPROACH TO TEXTURE ANALYSIS

S. Y. Lu
K. S. Fu

School of Electrical Engineering
Purdue University
West Lafayette, Indiana 47907

TR-EE 77-14

February 1977

This work was supported by the AFOSR Grant 74-2661.

AIR FORCE OFFICE OF SCIENTIFIC RESEARCH (AFSC)
NOTICE OF TRANSMITTAL TO DDC
This technical report has been reviewed and is approved for public release IAW AFR 190-12 (7b).
Distribution is unlimited.
A. D. BLOSE
Technical Information Officer

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

(19) REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

(18) 1. REPORT NUMBER: AFOSR - TR - 77 - 0382

2. GOVT ACCESSION NO.

3. RECIPIENT'S CATALOG NUMBER

(6) 4. TITLE (and Subtitle): A SYNTACTIC APPROACH TO TEXTURE ANALYSIS.

(9) 5. TYPE OF REPORT & PERIOD COVERED: Interim rept.

6. PERFORMING ORG. REPORT NUMBER

(10) 7. AUTHOR(s): S. Y. Lu, K. S. Fu

8. CONTRACT OR GRANT NUMBER(s): AFOSR 74-2661

9. PERFORMING ORGANIZATION NAME AND ADDRESS: Purdue University, School of Electrical Engineering, West Lafayette, Indian 47907

10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS: 61102F, 2304/A2

11. CONTROLLING OFFICE NAME AND ADDRESS: Air Force Office of Scientific Research/NM, Bolling AFB DC 20332

(11) 12. REPORT DATE: Feb 77

13. NUMBER OF PAGES: 101

14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office): (14) TR-EE 77-14

(12) 102p.

15. SECURITY CLASS. (of this report): UNCLASSIFIED

15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

(15) ✓ AF-AFOSR-2661-74

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

(16) 2304 (17) A2

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

A syntactic model for the generation of structured textures and for the discrimination of textures is proposed. A texture pattern is first divided into fixed-size windows. Windows belonging to the same texture pattern are then characterized by a tree grammar. This tree grammar is used for synthesis as well as discrimination. If it is considered necessary, distortion or noise is introduced by using stochastic tree grammars. Finally, a set of error-correcting tree automata is used as a texture discriminator. Illustrative examples for texture synthesis and discrimination are presented.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

292000

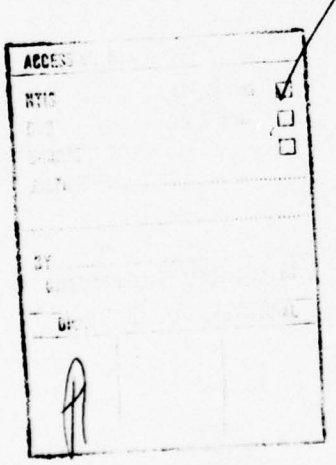
UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

A SYNTACTIC APPROACH TO TEXTURE ANALYSIS[†]

S. Y. Lu and K. S. Fu
School of Electrical Engineering
Purdue University
West Lafayette, Indiana 47907

TR-EE 77-14
February, 1977



[†]This work was supported by the AFOSR Grant 74-2661.

A SYNTACTIC APPROACH TO TEXTURE ANALYSIS[†]

S. Y. Lu and K. S. Fu
School of Electrical Engineering
Purdue University
West Lafayette, Indiana 47907

Abstract

A syntactic model for the generation of structured textures and for the discrimination of textures is proposed. A texture pattern is first divided into fixed-size windows. Windows belonging to the same texture pattern are then characterized by a tree grammar. This tree grammar is used for synthesis as well as discrimination. If it is considered necessary, distortion or noise is introduced by using stochastic tree grammars. Finally, a set of error-correcting tree automata is used as a texture discriminator. Illustrative examples for texture synthesis and discrimination are presented.

[†]This work was supported by the AFOSR Grant 74-2661.

1. Introduction

Research on texture analysis--modeling, synthesis, classification, and discrimination--has received increasing attention in recent years [1]. Texture is a term for the quality of a surface. The feature that dominates a texture scene is the repetitive or quasi-repetitive pattern. Texture information is valuable in scene segmentation, especially in those cases in which the contrast between the object to be observed and the background is poor. A survey of research in this area can be found in Zucker [2]. Applications of texture analysis include terrain classification [3, 4], radiographic image interpretation [5, 6], microscopic cell image analysis [7-9], and many others.

Most of the previous research has concentrated on the statistical approach [3-12]. In this approach, statistical properties are calculated from a set of local measurements taken from the pattern. Weszka, Dyer, and Rosenfeld [4] give a comparative study of several frequently used features for texture classification.

An alternative approach to the statistical one for texture analysis is the structural approach [1]. A texture is considered to be defined by subpatterns which occur repeatedly according to a set of well-defined placement rules within the overall pattern. Furthermore, the subpattern itself is made of structured elements. Compared with the statistical approach, the structural approach appears to be easier to interpret.

Zucker [2] proposed the idea of texture modeling in terms of an ideal texture and its transformations. According to Zucker, an ideal texture is a deep, unobservable, highly-structured prefect pattern in which local primitives (fundamental building blocks) are extended into a global structure such as regular tesselation. He believes that transformation rules can be defined from a representation of an ideal texture to that of a natural texture. In our work, we shall propose a tree grammar which defines such rules.

Carlucci [13] has formulated a system called "texture language" for the description of some simple repetitive subpatterns such as polygons or open polygonals. Texture patterns are treated as graphs with the basic elements in the texture language representation being lines and vertices. The structure of a subpattern is then represented as a tree. From a practical point of view, Carlucci's system may encounter difficulties during preprocessing, such as difficulty in the extraction of lines and vertices in a texture region.

We shall propose a texture model based on the structural approach. In this model, a texture pattern is divided into fixed-size windows. Repetition of subpatterns or a portion of a subpattern may appear in a window. For all cases, we shall treat a windowed pattern as a subpattern. A tree grammar is then used to characterize windowed patterns of the same class. This model can be used for texture synthesis as well as discrimination. Since the windowed patterns are also a part of the global structure of the texture, a higher level of syntax rules can also be constructed for the arrangement of windowed patterns.

Definitions and notations that appear in the subsequent sections are those used in the literature [14-17] and are briefly reviewed here.

Definition 1. A grammar $G_t = (V, r, P, S)$ over $<\Sigma, r>$ is a tree grammar in expansive form

where V is a set of terminal and nonterminal symbols,

Σ is a set of terminal symbols,

$r: \Sigma \rightarrow N$ where N is the set of non-negative integers, is the rank associated with symbols in Σ,

S is the start symbol,

and P is a set of production rules in the form of

$$X_0 \rightarrow \begin{array}{c} \diagup \; x \; \diagdown \\ X_1 \cdots X_{r(x)} \end{array} \qquad \text{or } X_0 \rightarrow x \qquad \text{where } x \in \Sigma$$

and $X_0, X_1, \ldots, X_{r(x)} \in V-\Sigma$

Definition 2. A stochastic tree grammar $G_S = (V, r, P, S)$ over $<\Sigma, r>$ is expansive if and only if each rule in P is of the form

$$X_0 \xrightarrow{P_{0i}} \overset{x}{\diagup\ \diagdown}_{X_1 \cdots X_{r(x)}} \quad \text{or} \quad X_0 \xrightarrow{P_{0i}} x$$

where $x \in \Sigma$, $X_0, X_1, \ldots, X_{r(x)} \in V-\Sigma$.

Definition 3. Let T_Σ^D be the set of all trees of the same structure D with nodes labeled by symbols in Σ. A mapping S: $T_\Sigma^D \to T_\Sigma^D$ is called substitution transformation. We shall write $\alpha \vert\xrightarrow{S_{a/x}} \beta$ iff $\alpha, \beta \in T_\Sigma^D$, a is a node in D and β is the result of replacing the label on node a of tree α by terminal symbol x.

Definition 4. The distance between two trees α, β in T_Σ^D, $d(\alpha, \beta)$ is defined as the smallest number of transformations required to derive β from α. Let L be a tree language. The distance between a tree β and L, $d(L, \beta)$ is defined as

$$d(L, \beta) = \min_{\alpha} \{d(\alpha, \beta) \mid \alpha \in L\}$$

2. A Syntactic Model for Texture Analysis

We propose the following syntactic model for texture analysis: its primitive, window, tree representation, and tree grammar being described here.

2.1. The Primitive

We choose a single pixel with different gray levels to be the pattern primitive. For a picture of ℓ gray levels, we have ℓ different primitives.

2.2. The Window

From the structural point of view, texture is the placement of structured subpatterns. (See Fig. 1(b)). However, in a natural scene, the exact boundary of a subpattern is usually vague and unidentifiable. Often, subpatterns of the same texture appear in various sizes, shapes, or brightness. In some cases, there even exists a situation in which there are no well-defined subpatterns. For examples of this see Figs. 1(c) and 1(d). In addition,

the placement of subpatterns can be irregular and distorted such as shown in Fig. 1(a). Nevertheless, a small subframe of the overall texture pattern does maintain some of the characteristics of the texture. To make the syntactic approach practically feasible, pictures are divided into fixed-size windows. A grammar is then used to characterize the windowed pattern of the given texture. Assuming that the window is of size k x k, there are ℓ^{k^2} possible patterns. The set of all the windowed patterns of a particular texture is a subset of the ℓ^{k^2} patterns. Consequently, a high-dimensional regular grammar, for example, a tree grammar [15], is suitable for the characterization of texture patterns.

2.3. The Tree Representation

Before we infer the tree grammar for a texture pattern, each windowed pattern is first transformed into a tree representation. Each node on the tree representation corresponds to a pixel in the k x k window. Hence, a pattern primitive becomes the assigned label to its corresponding node. For implementation, a tree structure can be arbitrarily chosen, but is then fixed for all windows during the process. That is, for all tree representations, the tree structure is the same, but node labels are different. Two convenient tree structures are suggested in Figs. 2(a) and 2(b) where the window size is 9 x 9. We shall refer to them as Structure A and Structure B, respectively. Clearly, a different choice of tree structure will result in a different tree representation for the same window. This choice will influence the complexity as well as the effectiveness of the inferred tree grammar.

Example 2.1. The pattern shown in Fig. 3(a) has the tree representation shown in Fig. 3(b) if Structure A is used.

2.4. The Tree Grammar

Example 2.2. The following tree grammar G_1 will generate the tree representation shown in Fig. 3(b):

$G_1 = (V_1, r, P_1, A_1)$ over $<\Sigma, r>$

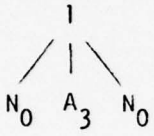
$V_1 = \{A_1, A_2, A_3, A_4, A_5, A_6, N_0, V_0, 1, 0\}$

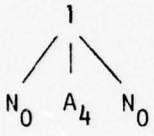
$\Sigma = \{$  ,  $\}$

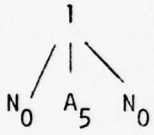
$r = \{0, 1, 2, 3\}$

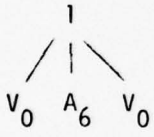
P_1:

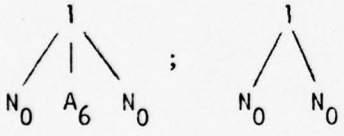
$A_1 \rightarrow$ 

$A_2 \rightarrow$ 

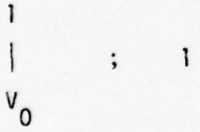
$A_3 \rightarrow$ 

$A_4 \rightarrow$ 

$A_5 \rightarrow$ 

$A_6 \rightarrow$ 

$N_0 \rightarrow$ 

$V_0 \rightarrow$ 

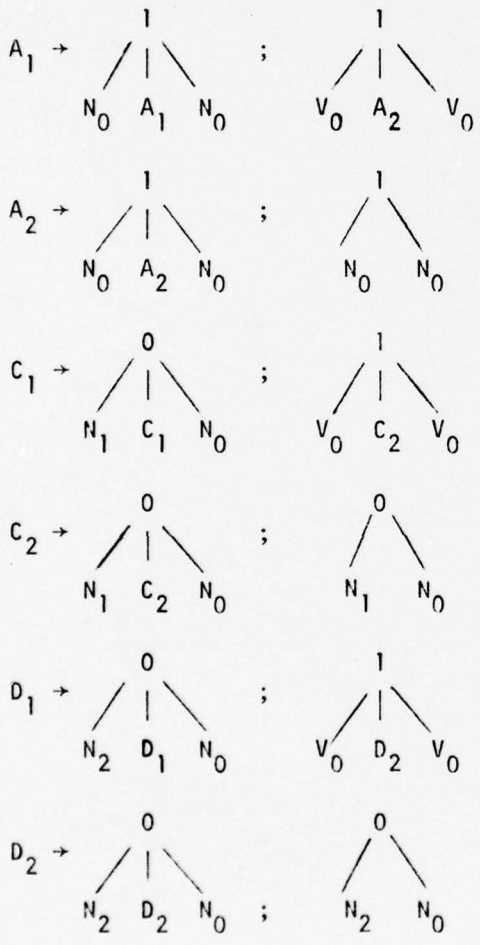
Example 2.3. Grammar G_1 in Example 2.2 will only accept patterns of a cross in the middle of the 9 x 9 window such as the texture pattern shown in Fig. 4. In a natural texture, we will most likely have some distortions of the prefect pattern such as the pattern shown in Fig. 5. Grammar G_2 will generate patterns having a shifting of the cross in Fig. 3(a) anywhere within the window.

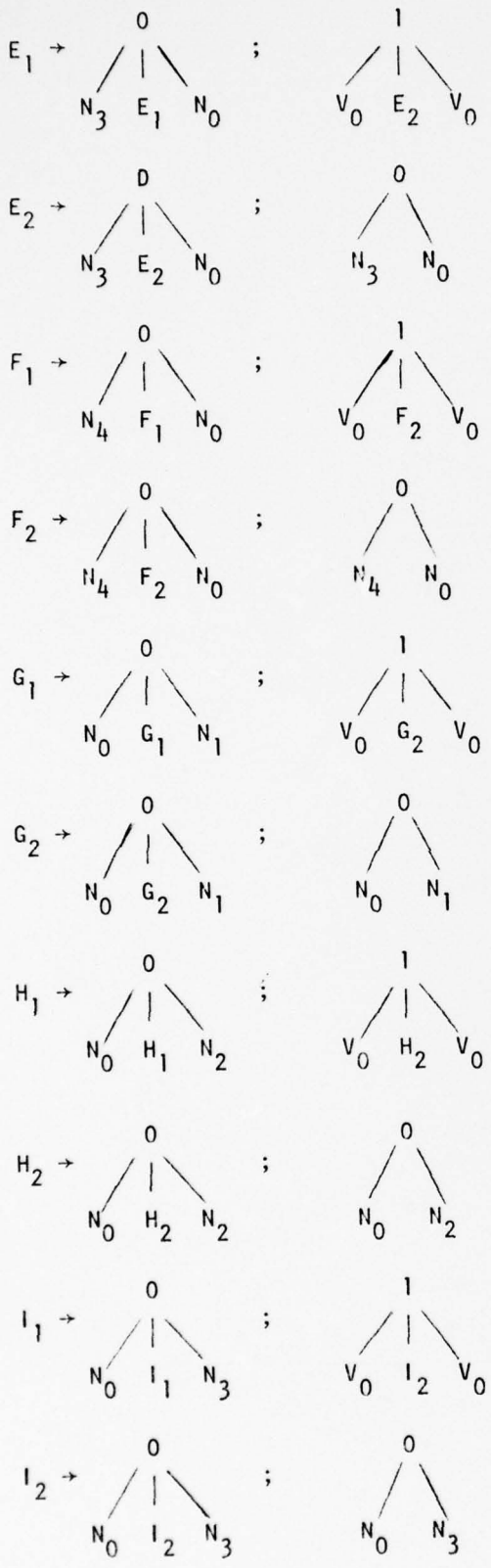
$G_2 = (V_2, r, P_2, S_2)$ over $<\Sigma, r>$

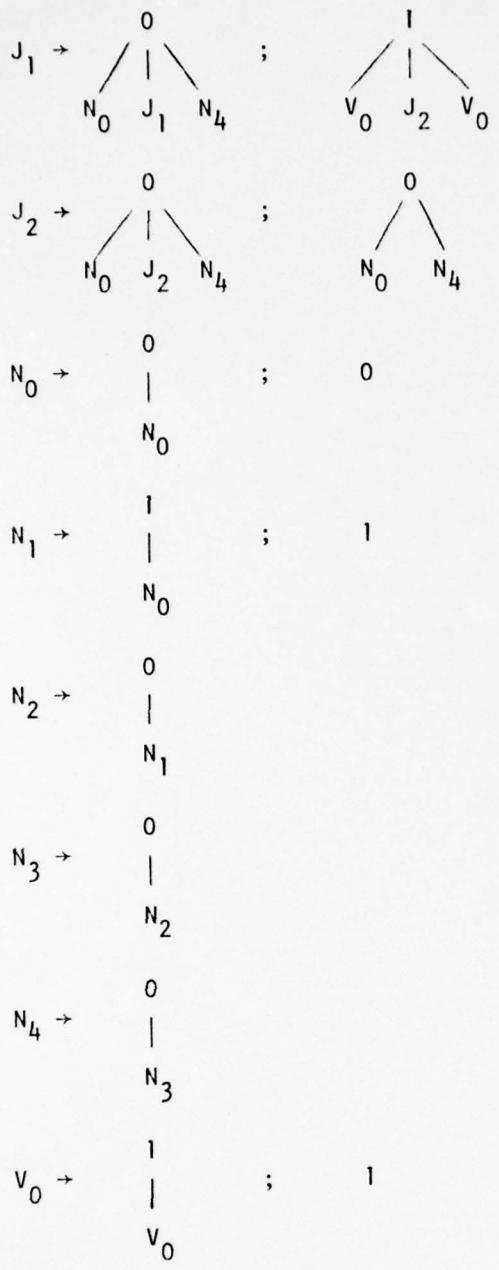
$V_2 = \{A_1, A_2, C_1, C_2, D_1, D_2, E_1, E_2, F_1, F_2, G_1, G_2, H_1, H_2, I_1, I_2, J_1, J_2,$
$N_0, N_1, N_2, N_3, N_4, V_0, 1, 0\}$

$S_2 = \{A_1, C_1, D_1, E_1, F_1, G_1, H_1, I_1, J_1\}$

P_2:




E1 → 0 N3 E1 N0 ; 1 V0 E2 V0
E2 → D N3 E2 N0 ; 0 N3 N0
F1 → 0 N4 F1 N0 ; 1 V0 F2 V0
F2 → 0 N4 F2 N0 ; 0 N4 N0
G1 → 0 N0 G1 N1 ; 1 V0 G2 V0
G2 → 0 N0 G2 N1 ; 0 N0 N1
H1 → 0 N0 H1 N2 ; 1 V0 H2 V0
H2 → 0 N0 H2 N2 ; 0 N0 N2
I1 → 0 N0 I1 N3 ; 1 V0 I2 V0
I2 → 0 N0 I2 N3 ; 0 N0 N3



The distorted pattern shown in Fig. 5 can be accepted by G_2.

3. Illustrative Examples of Texture Synthesis

In Section 2, a syntactic model is presented for describing windowed patterns. The global structure of the overall texture pattern depends on the arrangement of windowed patterns. In Example 2.3, we constructed the grammar G_2 for the acceptance

of the texture pattern shown in Fig. 5. However, when G_2 is used for generation, numerous patterns might be produced, one of which is shown in Fig. 6. Therefore, in order to preserve the coherence between windows, a set of higher level syntax rules is necessary in which the windowed pattern is treated as a primitive, and the overall texture can be represented as a tree which decides the placement of windowed patterns.

In this section, we will illustrate the synthesis of patterns D22, D34, D38, and D68 from Brodatz's Textures [18]. Figs. 1(a), (b), (c), and (d) are digitized pictures with resolutions of 400μ, 400μ, 100μ, and 400μ, respectively, of the above four patterns. For simplicity, we use only two primitives: black as primitive "1," and white as primitive "0." By setting a threshold for gray levels in Fig. 1, we obtain four binary pictures, Fig. 7(a), (b), (c), and (d) for patterns D22, D34, D38, and D68, respectively.

3.1. Regular Tesselation

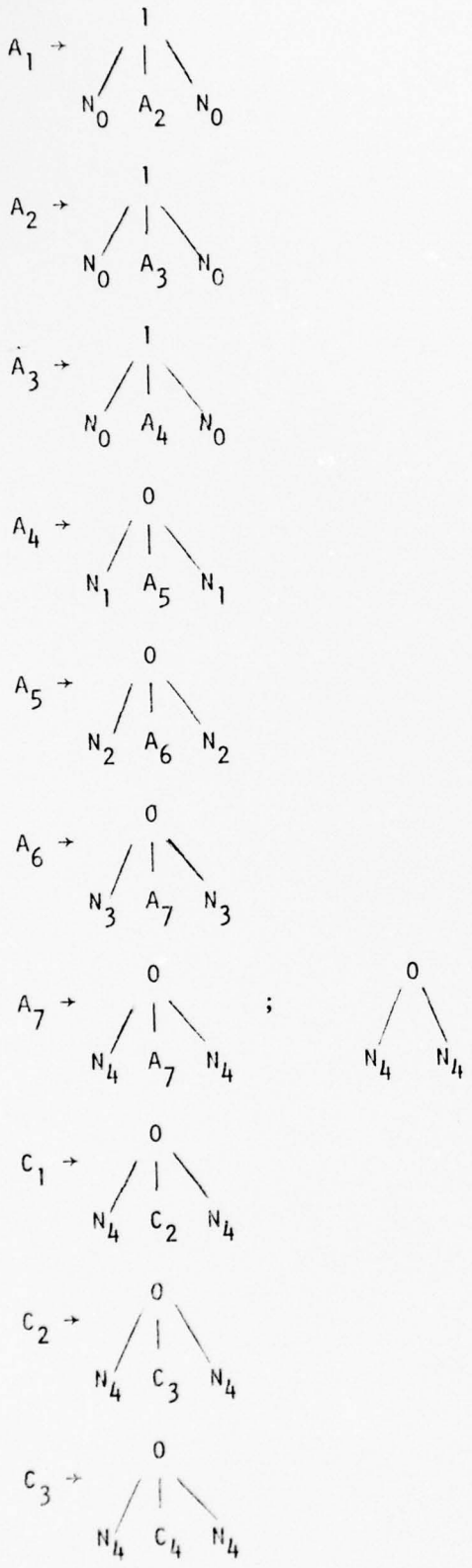
The texture pattern D34 is a hexagonally tesselated pattern which is nearly what Zucker called "ideal texture."

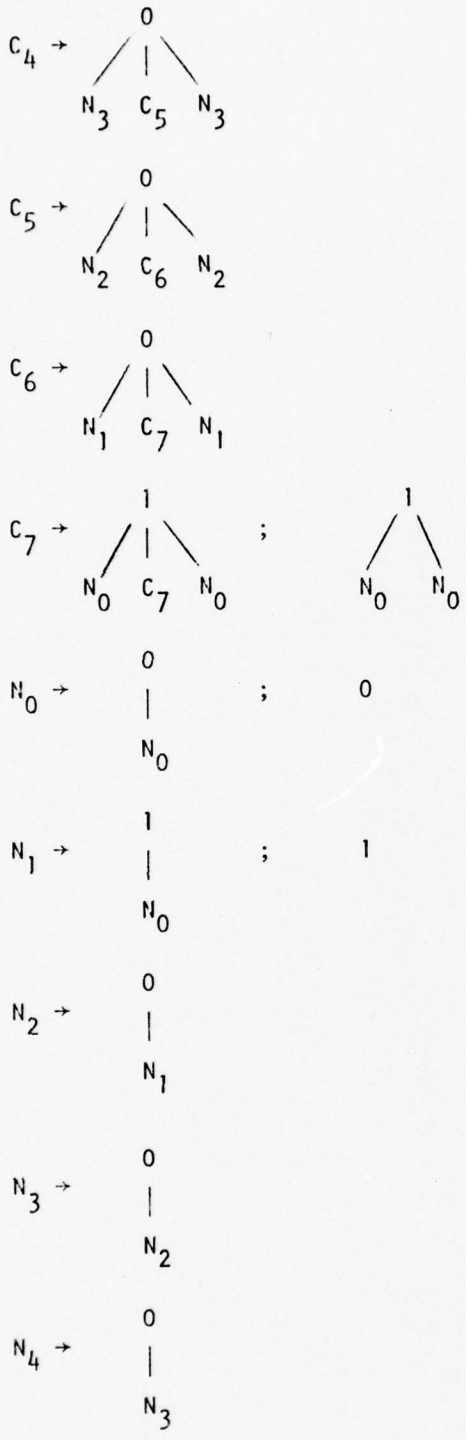
Example 3.1. A synthesis of hexagonal tesselation is as follows: Assume that we have two windowed patterns; namely, A_1 and C_1 shown in Fig. 8(a) and (b), respectively, with window-size 9 x 9. Tree grammar G_3 generates the tree representations, A_1 and C_1, using Structure A described in 2.3. Tree grammar G_3' generates the placement rule for A_1 and C_1. The placement rule is given in Structure B. G_3 and G_3' are given as follows:

$$G_3 = (V_3, r, P_3, \{A_1, C_1\}) \text{ over } <\Sigma, r>$$

$$V_3 = \{A_1, A_2, A_3, A_4, A_5, A_6, A_7, C_1, C_2, C_3, C_4, C_5, C_6, C_7, N_1, N_2, N_3, N_4, N_0, 1, 0\}$$

P_3:


A1 → 1 N0 A2 N0
A2 → 1 N0 A3 N0
A3 → 1 N0 A4 N0
A4 → 0 N1 A5 N1
A5 → 0 N2 A6 N2
A6 → 0 N3 A7 N3
A7 → 0 N4 A7 N4 ; 0 N4 N4
C1 → 0 N4 C2 N4
C2 → 0 N4 C3 N4
C3 → 0 N4 C4 N4


C4 → 0 N3 C5 N3
C5 → 0 N2 C6 N2
C6 → 0 N1 C7 N1
C7 → 1 N0 C7 N0 ; 1 N0 N0
N0 → 0 N0 ; 0
N1 → 1 N0 ; 1
N2 → 0 N1
N3 → 0 N2
N4 → 0 N3

$G_3' = (V_3', r', P_3', X)$ over $<\Sigma', r'>$

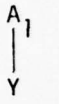
$V_3' = \{X, Y, Z, A_1, C_1\}$

$\Sigma' = \{A_1, C_1\}$, $r' = \{0, 1, 2\}$

P_3':

$X \rightarrow$  ; 

$Y \rightarrow$  ; C_1

$Z \rightarrow$  ; A_1

The generating procedure using the two-level syntax rules, G_3 and G_3' is illustrated in Fig. 9. The generated pattern is shown in Fig. 10.

Example 3.2. In Example 3.1, we assumed that the window size matched the size of the hexagonal subpattern. However, the 9 x 9 window in Example 3.1 does not perfectly match the pattern D34 in Fig. 7(b). An improvement is shown in Fig. 11(b) where the hexagons have a similar size to that of pattern D34. In Fig. 11(b), window frames have been drawn in so that the windowed patterns and their repetitive order can be shown clearly. There are 20 different windowed patterns within each heavily-lined area in a 4 x 5 arrangement. The larger pattern, made up of the 20 windows, also repeats itself. The grammar G_4 that generates the 20 windowed patterns, is given on the left-hand side of Appendix A-1. Fig. 11(a) shows the placement rule in Structure 8 for the pattern in Fig. 11(b). The symbol in each cell of Fig. 11(a) belongs to the set of starting symbols in grammar G_4. From each starting symbol, the corresponding windowed pattern of Fig. 11(b) can be generated. The grammar that generates the placement rule is also given in Appendix A-1 as grammar G_4'.

Example 3.3. The uneven brightness in pattern D34, e.g., darker for horizontal lines and lighter for diagonal lines, can be simulated by using a stochastic grammar. Fig. 12 is the resulting pattern from using stochastic grammar G_{S_4} in the generation of the "noisy version." The grammar G_{S_4} is given on the right-hand side in Appendix A-1.

In this subsection, three examples were given: (1) to illustrate the synthesis of an ideal texture for a matching sized window and subpattern, (2) for an unmatched size window and subpattern, and (3) for a noisy version. However, the noisy version described in Example 3.3 is the result of local noise. In the following subsection, we shall describe the synthesis of a global structure-distorted texture pattern.

3.2. Irregular Tesselation

Let us examine pattern D22 in Fig. 7(a). We may consider that pattern D22 is the result of twisting the regular tesselation of an ideal texture such as the pattern shown in Fig. 13. From a single window, the trend of distortion cannot be fully detected. For texture synthesis, such a global distortion can be treated as a problem of the placement of windowed patterns.

Example 3.4. The regular tesselated pattern shown in Fig. 13 is composed of two basic patterns shown in Fig. 14(a) and (b). A distorted tesselation can result from shifting a series of basic patterns in one direction. Let us use the set of patterns resulting from shifting a basic pattern as the set of primitives. There will be 81 such windowed pattern primitives. We shall refer to them simply as primitives in this example. Fig. 15 shows several of them. Each primitive is given a name of two symbols. "X_i," where $X \in \{A, B, C, D, E, F, G, H, I\}$ $i \in \{1, 2, \ldots, 9\}$. Starting from X_i, the pattern resulting from shifting one column to the left will be named X_{i+1}, and the pattern resulting from shifting one row up will be named Y_i. Grammar G_5 in Appendix A-2 is constructed for the generation of the 81 primitives.

Several synthesis results are given in Fig. 16(a), (b), and (c). Tree representations using Structure B that decide the placement of windowed patterns are shown at the left-hand side of each pattern.

Using the same idea as that in Example 3.3, a stochastic grammar can be used to add local distortions. We can also construct a grammar for the placement of windowed patterns for certain types of structure distortion. For example, a twisted upward or downward pattern, or an insertion of an extraneous row of subpatterns, etc.

3.3. Random Pattern

The texture pattern D38 and D68 in Fig. 7(c) and (d) show a higher degree of randomness than D22 and D34. No clear tesselation or subpattern exists in the pattern.

Example 3.5. The water waves in pattern D38 can be described as a belt extending in the horizontal direction, varying in width and twisting upward or downward. Assuming that, at most, one belt can appear in a window, we shall use Structure A for tree representation and a stochastic tree grammar G_6 to describe such patterns. Each production rule in G_6, the left-hand side nonterminal, is the present state and the right-hand side generates the width and the position of the belt that the present state represented, as well as the next state. Fig. 17 illustrates this generation process. The grammar G_6 is given in Appendix B-1. G_6 is also the discrimination grammar for pattern D38 which will be discussed in Section 4. The production rules associated with zero probability are unused rules during pattern generation. They are added for pattern discrimination. The probabilities associated with the production rules in G_6 are arbitrarily assigned. By varying the assignment of probabilities, patterns with a different degree of brightness and fluctuation can be generated. Some resulting patterns are shown in Fig. 18.

Example 3.6. The texture pattern of D68, the wood grain pattern, consists of long vertical lines. It is particularly convenient for syntactic description when

Structure A is used for tree representation. The subpattern (vertical line) and its repetition can be fully characterized by the stochastic grammar G_7. Therefore, there is no need to generate the overall pattern window by window. The grammar G_7 is given as follows:

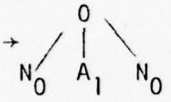
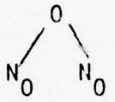
$G_7 = (V_7, r, P_7, A_1)$ over $<\Sigma, r>$

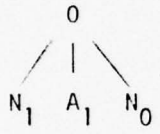
$V_7 = \{A_1, N_0, N_1, 0, 1\}$

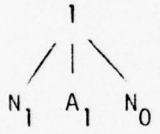
$r = \{0, 1, 2, 3\}$

$\Sigma = \{0, 1\}$

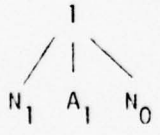
P_7:

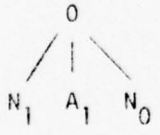
$A_1 \rightarrow$  , 0.5 ;  , 0.05 ;

 , 0.09 ;

 , 0.09 ;

 , 0.09 ;

 , 0.09 ;

 , 0.09 ;

$$N_0 \rightarrow \begin{array}{l} \begin{matrix} 0 \\ | \\ N_0 \end{matrix} \quad , \ 0.90 \quad ; \\ \begin{matrix} 0 \\ | \\ N_1 \end{matrix} \quad , \ 0.05 \quad ; \\ 0 \quad , \ 0.05 \quad ; \end{array}$$

$$N_1 \rightarrow \begin{array}{l} \begin{matrix} 1 \\ | \\ N_1 \end{matrix} \quad , \ 0.85 \quad ; \\ \begin{matrix} 1 \\ | \\ N_0 \end{matrix} \quad , \ 0.10 \quad ; \\ 1 \quad , \ 0.05 \end{array}$$

The density of grains (vertical lines) depends on the probabilities associated with production rules. Pictures in Fig. 19 are generated from G_7 using different probability assignments.

4. Texture Discrimination

The proposed texture model can also be used for texture discrimination. In Section 3, we illustrated how a texture pattern was generated window-by-window. The construction of a grammar in modeling the variation of size, shape, and brightness, as well as noise and distortion was illustrated by examples. We also discussed in Section 2 that a pattern in a small subframe (window) maintains some of the characteristics of the overall texture. Under this assumption, we shall restrict the problem of texture discrimination to the recognition of windowed patterns only. Each picture is processed window-by-window.

4.1. Data Preparation

The pattern shown in Fig. 20 consists of patterns D22, D34, D38, and D68. There are 180 x 180 pixels with 128 gray levels. We shall use two primitives (two gray levels) for discrimination. The picture shown in Fig. 21 is obtained by setting a threshold at gray level 44. Window frames are drawn in in Fig. 21. The window size is 9 x 9.

4.2. Discrimination Grammars

The texture modeling grammars described in Section 3 are used for discrimination here. Let the grammar for pattern D22, D34, D38, and D68 be G_{22}, G_{34}, G_{38}, and G_{68}, respectively. From the viewpoint of discrimination, we would like to modify the grammar so that overlaps between $L(G_{22})$, $L(G_{34})$, $L(G_{38})$, and $L(G_{68})$ (languages generated from G_{22}, G_{34}, G_{38}, and G_{68}, respectively) will be as small as possible. Whereas, each language itself needs to be as general in characterizing each class of texture as possible. Grammar G_{22}, G_{34}, and G_{68} are given in Appendix B-2, B-3, and B-4, respectively. Grammar G_{38} is the nonstochastic version of grammar G_6 in Appendix B-1.

4.3. Error-Correcting Parsing

The conventional parser usually fails to recognize a "noisy" pattern. Although we have tried to construct the discrimination grammars to include as large a variety of patterns as possible, the uncertainty existing in a pattern is impossible to be fully characterized and predicted. An error-correcting parser can be used to improve the classification accuracy [19-22]. In particular, in this application, we shall use the "structure-preserved error-correcting tree automata (ECTA)" as the texture discriminator. The algorithm for ECTA is presented in [22].

4.4. Computation Result

The ECTA measures the distance between the input tree representation and the texture languages, $L(G_{22})$, $L(G_{34})$, $L(G_{38})$, and $L(G_{68})$ one by one. Then, the input pattern is classified to the texture class which has the minimum distance.

The result of texture discrimination for the picture in Fig. 21 is given in Fig. 22. There are 400 windows. Thirty of them are misrecognized. The misrecognition usually results from the unavoidable overlap between two languages or from the reduction of one language to decrease the overlap.

5. Remarks

In this paper, a syntactic approach for texture modeling is presented. The proposed approach appears to be attractive from the practical point of view. The preprocessing involves picture digitization only. The window operation stores a small subframe of the pattern in the main memory. Thus, the process is manageable by a small memory computer.

The most difficult part comes from the construction of an effective grammar. Since no sophisticated preprocessing is used, the linguistic representations are very sensitive to noise. In constructing a grammar, we would like to consider as many variations of the texture pattern as possible. On the other hand, we also need to keep the grammar as simple (as few nonterminals and production rules) as possible to save storage space. Such a compromise often results in a grammar that generates some excessive sentences, but excludes some possible distortions. That is one reason for the necessity of using an error-correcting parser for picture parsing in texture discrimination. The other reason is the uncertainty existing in the picture making the construction of a grammar difficult in order to fit all the possibilities of a texture class.

All the computational examples are programmed in Fortran IV on a PDP-11/45 computer with a 32K core memory. The ECTA we designed processes all the branches of a tree from the frontiers to the root in parallel, but it should be programmed in series on a general purpose computer. The process can certainly be speeded up by a specially designed processor.

Automatic grammatical inference procedures have been recently studied [23]. By combining an inference algorithm with the proposed discrimination procedure, an automation of the entire training and testing process is possible [24].

References

[1] Lipkin, B. S. and A. Rosenfeld, Eds., Picture Processing and Psychopictorics, New York: Academic Press, Inc., 1970, pp. 289-381.

[2] Zucker, S. W., "Toward a Model of Texture," Computer Graphics and Image Processing 5, 1976, pp. 190-202.

[3] Haralick, R. M., K. Shanmugam and I. Dinstein, "Textural Features for Image Classification," IEEE Trans on SMC, Vol. SMC-3, No. 6, Nov. 1973.

[4] Weszka, J. S., C. R. Dyer and A. Rosenfeld, "A Comparative Study of Texture Measures for Terrain Classification," IEEE Trans. on SMC, Vol. SMC-6, No. 4, April, 1976.

[5] Sutton, R. N. and E. L. Hall, "Texture Measures for Automatic Classification of Pulmonary Disease," IEEE Trans. on Computers, Vol. C-21, No. 7, July 1972.

[6] Conners, R. W. and C. A. Harlow, "Some Theoretical Considerations Concerning Texture Analysis of Radiographic Images," presented at the 1976 IEEE Conference on Decision and Control, Dec. 1-3, Clearwater Beach, FL.

[7] Mui, J. K., J. W. Bacus and K. S. Fu, "A Scene Segmentation Technique for Microscopic Cell Images," presented at the Third International Joint Conference on Pattern Recognition, Nov. 8-11, 1976, Caronado, CA.

[8] Young, I. T., "The Classification of White Blood Cells," IEEE Trans. on Biomed. Eng., Vol. BME-19, No. 4, pp. 291-298, July 1972.

[9] Bacus, J. W. and E. E. Gose, "Leukocyte Pattern Recognition," IEEE Trans. on SMC, Vol. SMC-2, No. 4, pp. 513-526, Sept. 1972.

[10] McCormick, B. H. and S. N. Jayaramamurthy, "Time Series Model for Texture Synthesis," Int. J. of Compt. and Inf. Sci., Vol. 3, No. 4, 1974.

[11] McCormick, B. H. and S. N. Jayaramamurthy, "A Decision Theory Method for the Analysis of Texture," Int. J. of Compt. and Inf. Sci., Vol. 4, No. 1, 1975.

[12] Schachter, B. J., A. Rosenfeld and L. S. Davis, "Random Mosaic Models for Textures," Computer Science Technical Report Series, University of Maryland, 1976.

[13] Carlucci, L., "A Formal System for Texture Languages," Pattern Recognition, Vol. 4, pp. 53-72, 1972.

[14] Brainerd, W. S., "Tree Generating Regular Systems," Inf. and Control, 14, pp. 217-231, 1969.

[15] Fu, K. S. and B. K. Bhargava, "Tree Systems for Syntactic Pattern Recognition IEEE Trans. on Computers, Vol. C-22, No. 12, pp. 1087-1099, Dec. 1973.

[16] Bhargave, B. K. and K. S. Fu, "Stochastic Tree Systems for Syntactic Pattern Recognition," Proceedings, Allerton Conference on Circuit and System Theory, 1974.

[17] Lu, S. Y. and K. S. Fu, "Error-Correcting Syntax Analysis for Tree Language," EE-TR 76-24, Purdue University, July 1976.

[18] Brodatz, P., Textures, Dover Publications, New York, 1966.

[19] Fung, L. W. and K. S. Fu, "Stochastic Syntactic Decoding for Pattern Classification," IEEE Trans. on Computers, Vol. C-24, No. 6, July, 1975.

[20] Thomason, M. G. and R. C. Gonzalez, "Syntactic Recognition of Imperfectly Specified Patterns," IEEE Trans. on Computers, January, 1975.

[21] Lu, S. Y. and K. S. Fu, "Stochastic Error-Correcting Syntax Analysis for Recognition of Noisy Patterns," TR-EE 76-9, Purdue University, March, 1976. Accepted for IEEE Trans. on Computers, 1977.

[22] Lu, S. Y. and K. S. Fu, "Structure - Preserved Error-Correcting Tree Automata for Syntactic Pattern Recognition," Proceedings of the 1976 IEEE Conference on Decision and Control, Dec. 1-3, Clearwater Beach, FL

[23] Brayer, J. M. and K. S. Fu, "A Note on K-Tail Method of Tree Grammar Inference," IEEE Trans. on SMC, April 1977.

[24] Fu, K. S. and S. Y. Lu, "A Clustering Procedure for Syntactic Patterns," to be presented at the 1977 IEEE-CS Workshop on Picture Data Description and Management, April 21-22, University of Illinois at Chicago Circle, Chicago, IL.

Appendix A. Synthesis Grammars for Netting and Reptile Skin

A-1. Grammar for Netting (pattern D34)

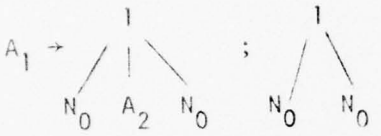
$G_4 = (V_4, r, P_4, S_4)$ over $<\Sigma, r>$ and $G_{S_4} = (V_4, r, P_{S_4}, S_4)$ over $<\Sigma, r>$

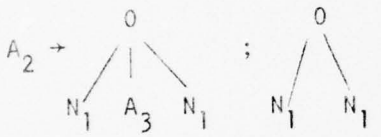
where $V_4 = \{A_{0,1,\ldots,9}, B_{0,1,\ldots,9}, C_{0,1,\ldots,9}, D_{0,1,\ldots,9}, E_{0,1,\ldots,9}, F_{0,1,\ldots,9}, N_{0,1,\ldots,6}, V_{0,1,\ldots 3}\} \cup \Sigma$.

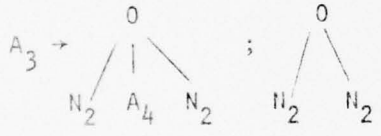
$S_4 = \{A_1, B_0, B_9, C_8, A_7, B_6, C_5, A_4, B_3, C_2, D_1, E_0, E_9, F_8, D_7, E_6, F_5, D_4, E_3, F_2\}$

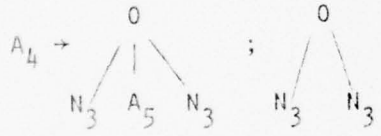
and

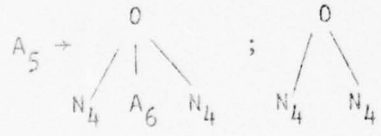
P_4:

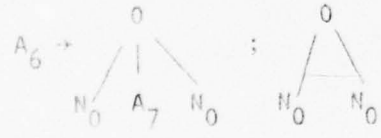




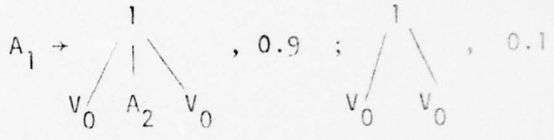


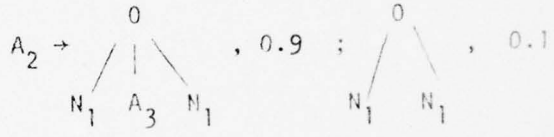


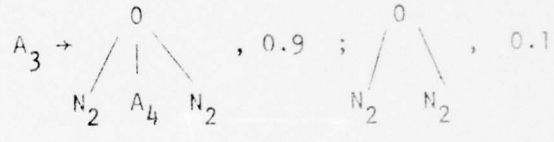


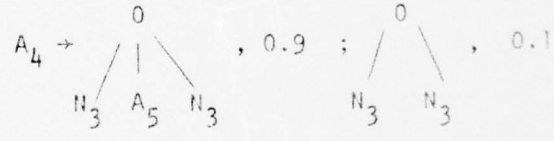


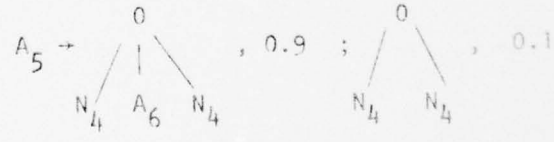
P_{S_4}:

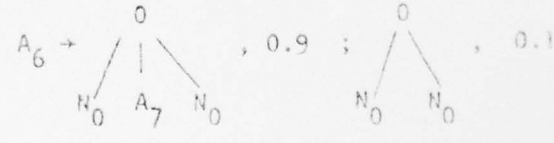


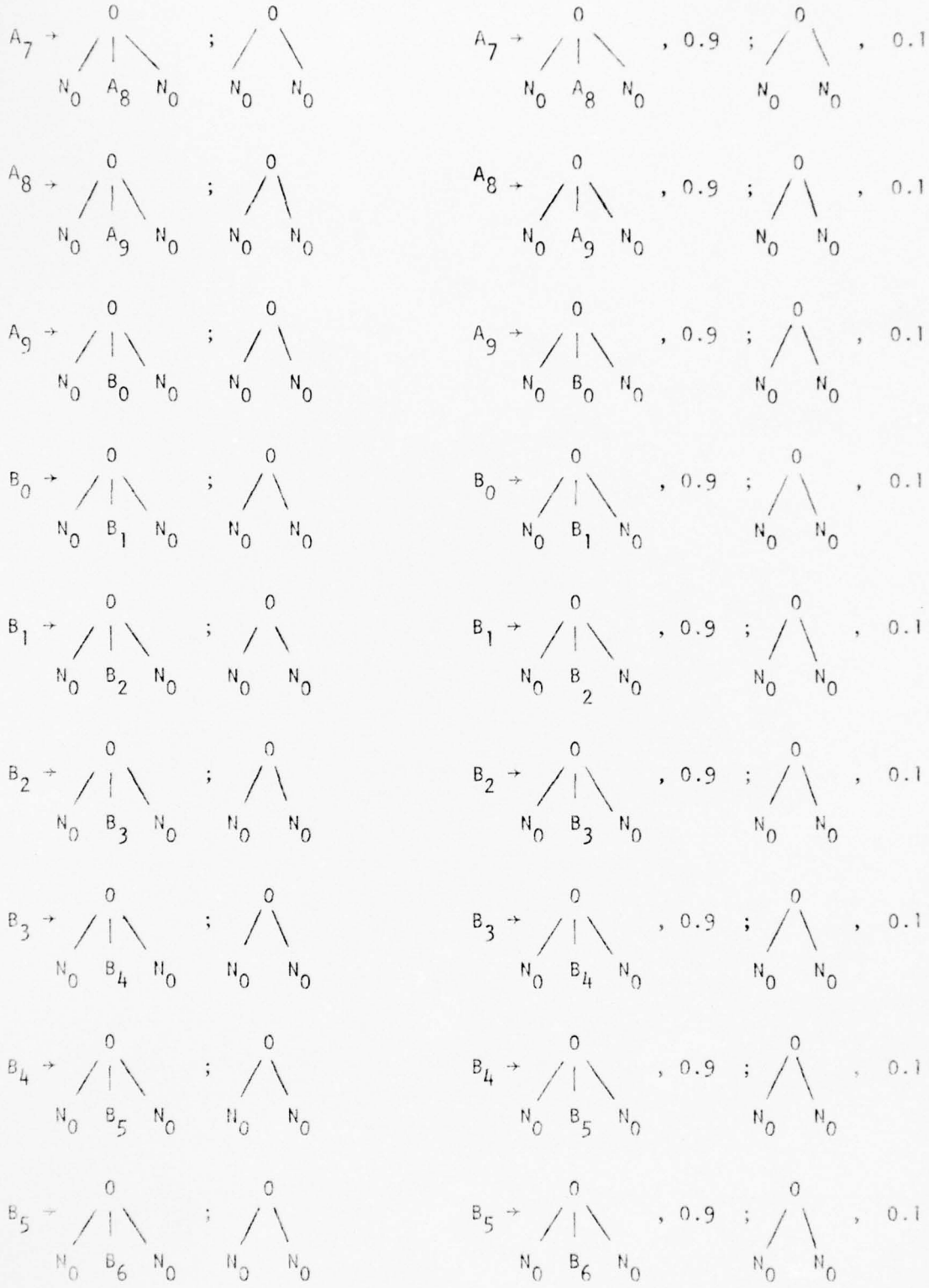


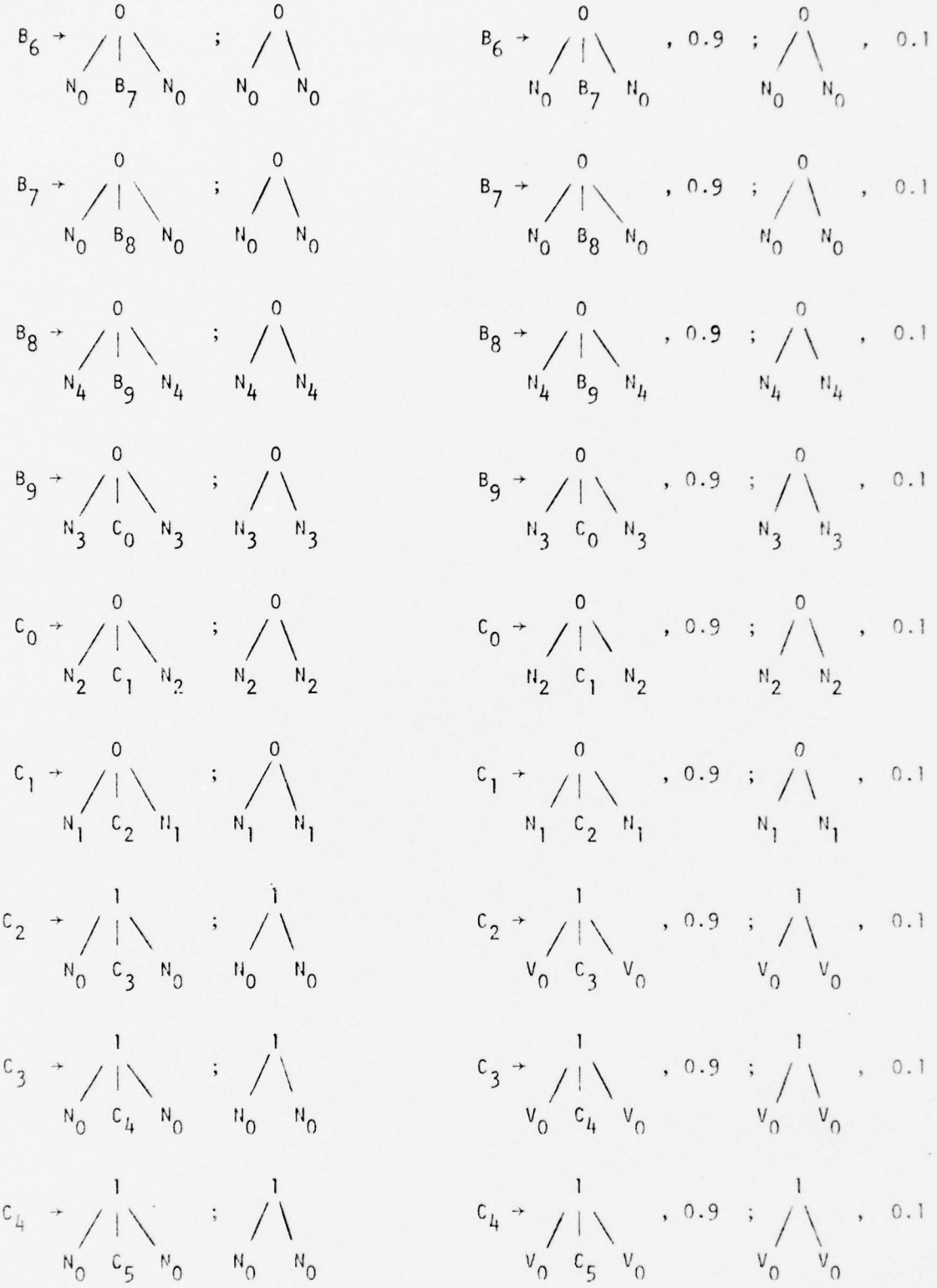


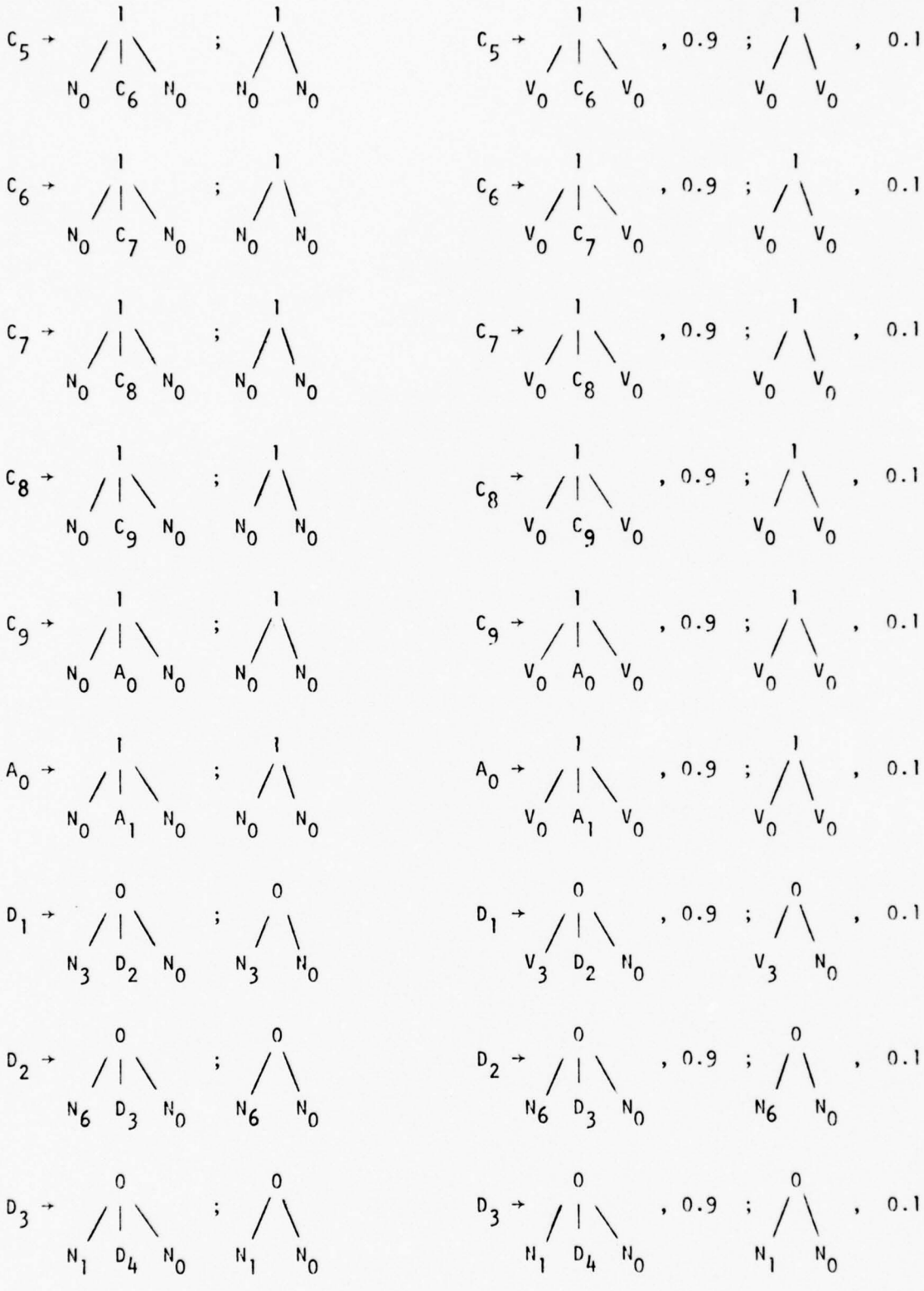


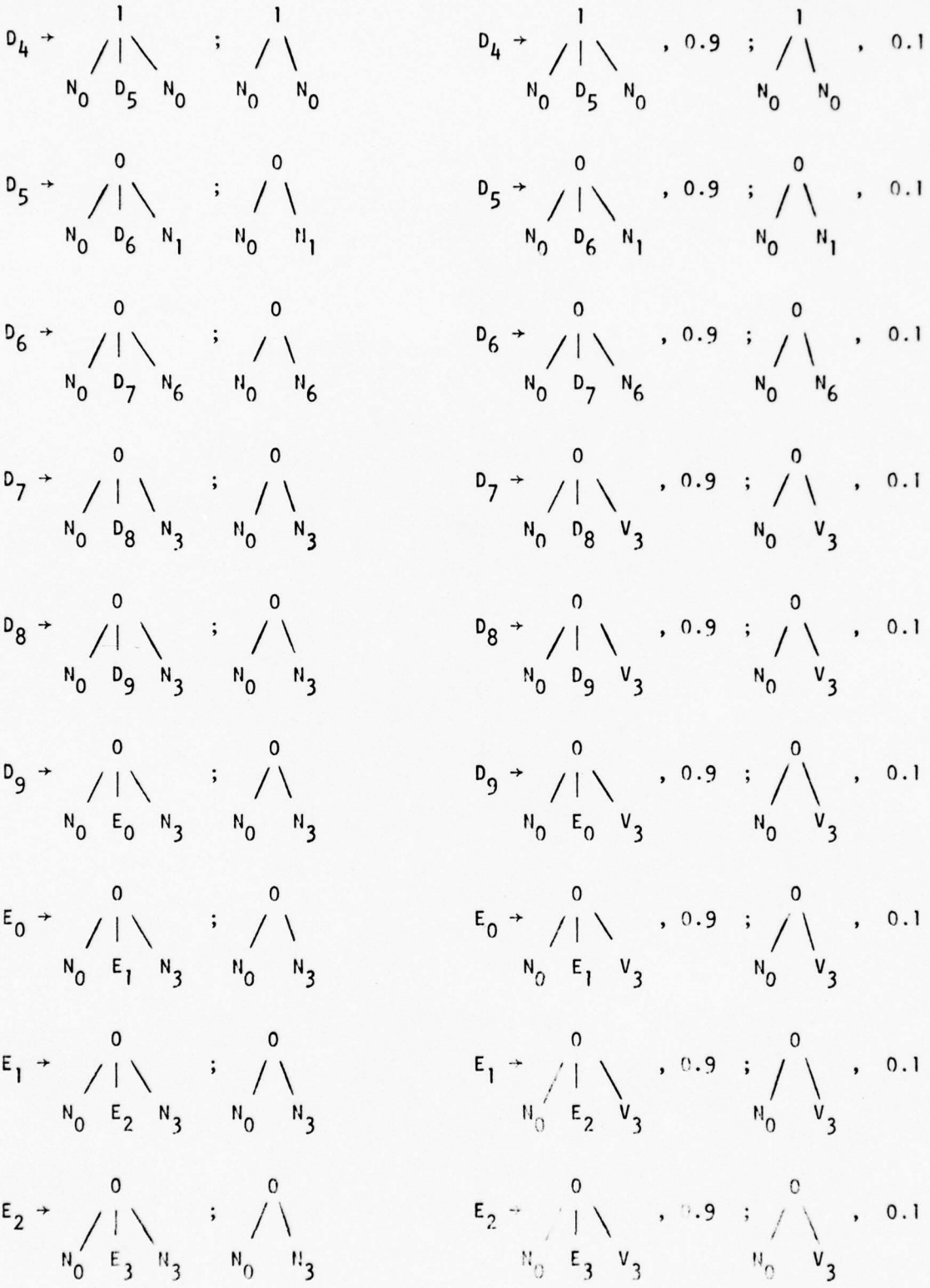


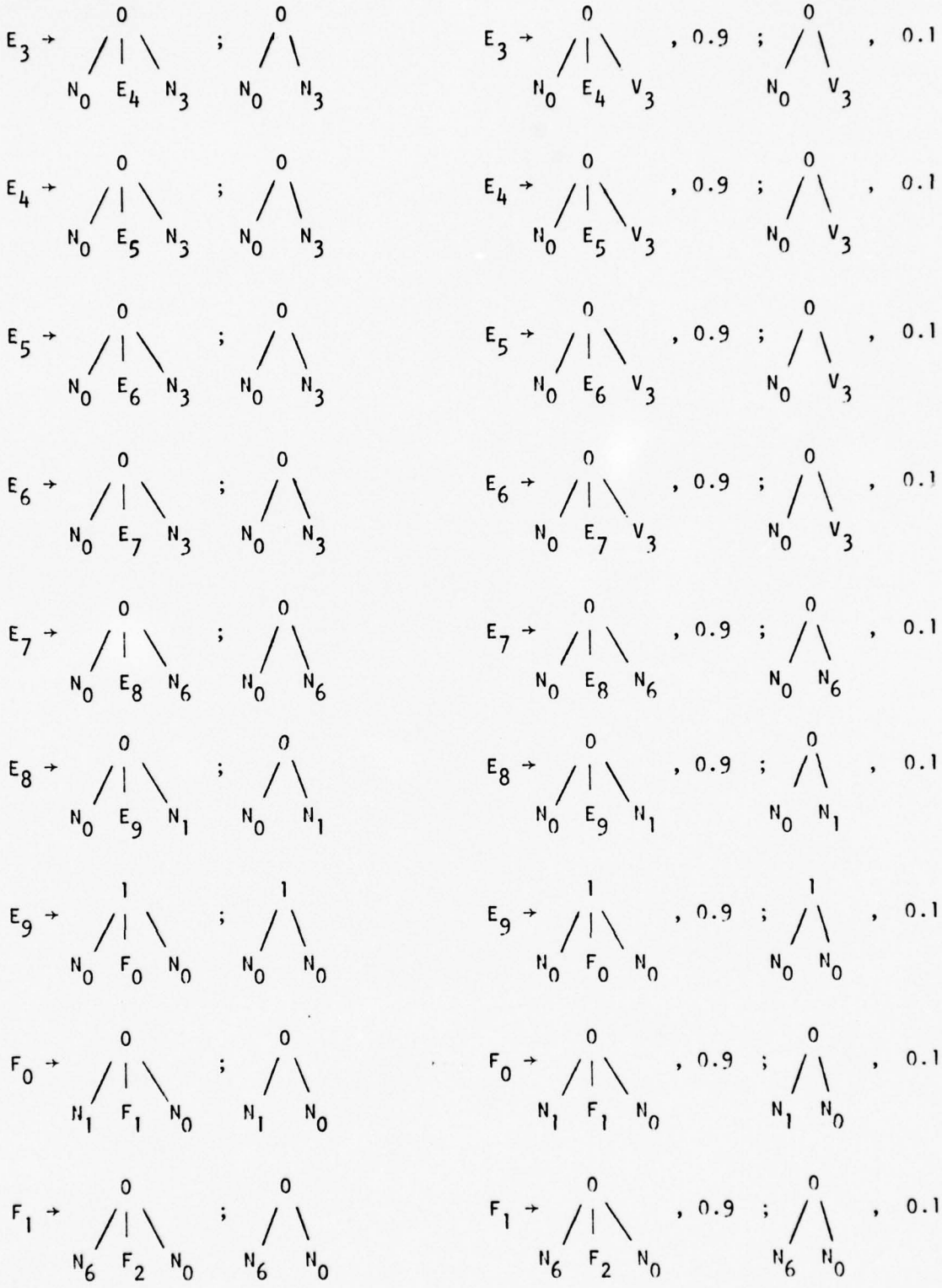


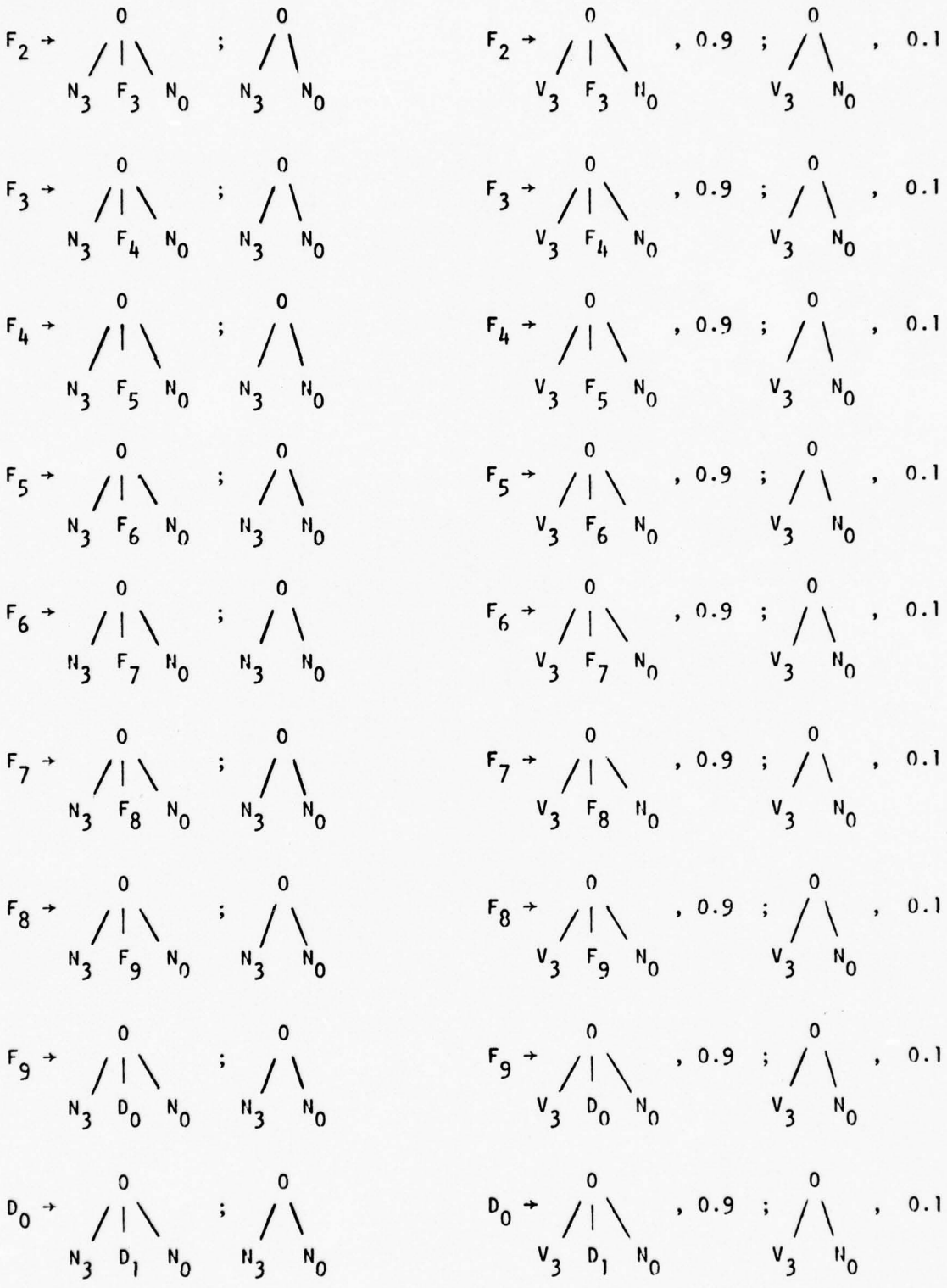

A7 → 0 ; 0 N0 A8 N0 N0 N0
A7 → 0 , 0.9 ; 0 , 0.1 N0 A8 N0 N0 N0
A8 → 0 ; 0 N0 A9 N0 N0 N0
A8 → 0 , 0.9 ; 0 , 0.1 N0 A9 N0 N0 N0
A9 → 0 ; 0 N0 B0 N0 N0 N0
A9 → 0 , 0.9 ; 0 , 0.1 N0 B0 N0 N0 N0
B0 → 0 ; 0 N0 B1 N0 N0 N0
B0 → 0 , 0.9 ; 0 , 0.1 N0 B1 N0 N0 N0
B1 → 0 ; 0 N0 B2 N0 N0 N0
B1 → 0 , 0.9 ; 0 , 0.1 N0 B2 N0 N0 N0
B2 → 0 ; 0 N0 B3 N0 N0 N0
B2 → 0 , 0.9 ; 0 , 0.1 N0 B3 N0 N0 N0
B3 → 0 ; 0 N0 B4 N0 N0 N0
B3 → 0 , 0.9 ; 0 , 0.1 N0 B4 N0 N0 N0
B4 → 0 ; 0 N0 B5 N0 N0 N0
B4 → 0 , 0.9 ; 0 , 0.1 N0 B5 N0 N0 N0
B5 → 0 ; 0 N0 B6 N0 N0 N0
B5 → 0 , 0.9 ; 0 , 0.1 N0 B6 N0 N0 N0


B6 → 0 N0 B7 N0 ; 0 N0 N0
B6 → 0 N0 B7 N0 , 0.9 ; 0 N0 N0 , 0.1
B7 → 0 N0 B8 N0 ; 0 N0 N0
B7 → 0 N0 B8 N0 , 0.9 ; 0 N0 N0 , 0.1
B8 → 0 N4 B9 N4 ; 0 N4 N4
B8 → 0 N4 B9 N4 , 0.9 ; 0 N4 N4 , 0.1
B9 → 0 N3 C0 N3 ; 0 N3 N3
B9 → 0 N3 C0 N3 , 0.9 ; 0 N3 N3 , 0.1
C0 → 0 N2 C1 N2 ; 0 N2 N2
C0 → 0 N2 C1 N2 , 0.9 ; 0 N2 N2 , 0.1
C1 → 0 N1 C2 N1 ; 0 N1 N1
C1 → 0 N1 C2 N1 , 0.9 ; 0 N1 N1 , 0.1
C2 → 1 N0 C3 N0 ; 1 N0 N0
C2 → 1 V0 C3 V0 , 0.9 ; 1 V0 V0 , 0.1
C3 → 1 N0 C4 N0 ; 1 N0 N0
C3 → 1 V0 C4 V0 , 0.9 ; 1 V0 V0 , 0.1
C4 → 1 N0 C5 N0 ; 1 N0 N0
C4 → 1 V0 C5 V0 , 0.9 ; 1 V0 V0 , 0.1


C5 → 1 N0 C6 N0 ; 1 N0 N0
C5 → 1 V0 C6 V0 , 0.9 ; 1 V0 V0 , 0.1
C6 → 1 N0 C7 N0 ; 1 N0 N0
C6 → 1 V0 C7 V0 , 0.9 ; 1 V0 V0 , 0.1
C7 → 1 N0 C8 N0 ; 1 N0 N0
C7 → 1 V0 C8 V0 , 0.9 ; 1 V0 V0 , 0.1
C8 → 1 N0 C9 N0 ; 1 N0 N0
C8 → 1 V0 C9 V0 , 0.9 ; 1 V0 V0 , 0.1
C9 → 1 N0 A0 N0 ; 1 N0 N0
C9 → 1 V0 A0 V0 , 0.9 ; 1 V0 V0 , 0.1
A0 → 1 N0 A1 N0 ; 1 N0 N0
A0 → 1 V0 A1 V0 , 0.9 ; 1 V0 V0 , 0.1
D1 → 0 N3 D2 N0 ; 0 N3 N0
D1 → 0 V3 D2 N0 , 0.9 ; 0 V3 N0 , 0.1
D2 → 0 N6 D3 N0 ; 0 N6 N0
D2 → 0 N6 D3 N0 , 0.9 ; 0 N6 N0 , 0.1
D3 → 0 N1 D4 N0 ; 0 N1 N0
D3 → 0 N1 D4 N0 , 0.9 ; 0 N1 N0 , 0.1


D4 → 1 N0 D5 N0 ; 1 N0 N0
D4 → 1 N0 D5 N0 , 0.9 ; 1 N0 N0 , 0.1
D5 → 0 N0 D6 N1 ; 0 N0 N1
D5 → 0 N0 D6 N1 , 0.9 ; 0 N0 N1 , 0.1
D6 → 0 N0 D7 N6 ; 0 N0 N6
D6 → 0 N0 D7 N6 , 0.9 ; 0 N0 N6 , 0.1
D7 → 0 N0 D8 N3 ; 0 N0 N3
D7 → 0 N0 D8 V3 , 0.9 ; 0 N0 V3 , 0.1
D8 → 0 N0 D9 N3 ; 0 N0 N3
D8 → 0 N0 D9 V3 , 0.9 ; 0 N0 V3 , 0.1
D9 → 0 N0 E0 N3 ; 0 N0 N3
D9 → 0 N0 E0 V3 , 0.9 ; 0 N0 V3 , 0.1
E0 → 0 N0 E1 N3 ; 0 N0 N3
E0 → 0 N0 E1 V3 , 0.9 ; 0 N0 V3 , 0.1
E1 → 0 N0 E2 N3 ; 0 N0 N3
E1 → 0 N0 E2 V3 , 0.9 ; 0 N0 V3 , 0.1
E2 → 0 N0 E3 N3 ; 0 N0 N3
E2 → 0 N0 E3 V3 , 0.9 ; 0 N0 V3 , 0.1


E3 → 0 N0 E4 N3 ; 0 N0 N3
E3 → 0 N0 E4 V3 , 0.9 ; 0 N0 V3 , 0.1
E4 → 0 N0 E5 N3 ; 0 N0 N3
E4 → 0 N0 E5 V3 , 0.9 ; 0 N0 V3 , 0.1
E5 → 0 N0 E6 N3 ; 0 N0 N3
E5 → 0 N0 E6 V3 , 0.9 ; 0 N0 V3 , 0.1
E6 → 0 N0 E7 N3 ; 0 N0 N3
E6 → 0 N0 E7 V3 , 0.9 ; 0 N0 V3 , 0.1
E7 → 0 N0 E8 N6 ; 0 N0 N6
E7 → 0 N0 E8 N6 , 0.9 ; 0 N0 N6 , 0.1
E8 → 0 N0 E9 N1 ; 0 N0 N1
E8 → 0 N0 E9 N1 , 0.9 ; 0 N0 N1 , 0.1
E9 → 1 N0 F0 N0 ; 1 N0 N0
E9 → 1 N0 F0 N0 , 0.9 ; 1 N0 N0 , 0.1
F0 → 0 N1 F1 N0 ; 0 N1 N0
F0 → 0 N1 F1 N0 , 0.9 ; 0 N1 N0 , 0.1
F1 → 0 N6 F2 N0 ; 0 N6 N0
F1 → 0 N6 F2 N0 , 0.9 ; 0 N6 N0 , 0.1


F2 → O (N3 F3 N0) ; O (N3 N0)
F2 → O (V3 F3 N0) , 0.9 ; O (V3 N0) , 0.1
F3 → O (N3 F4 N0) ; O (N3 N0)
F3 → O (V3 F4 N0) , 0.9 ; O (V3 N0) , 0.1
F4 → O (N3 F5 N0) ; O (N3 N0)
F4 → O (V3 F5 N0) , 0.9 ; O (V3 N0) , 0.1
F5 → O (N3 F6 N0) ; O (N3 N0)
F5 → O (V3 F6 N0) , 0.9 ; O (V3 N0) , 0.1
F6 → O (N3 F7 N0) ; O (N3 N0)
F6 → O (V3 F7 N0) , 0.9 ; O (V3 N0) , 0.1
F7 → O (N3 F8 N0) ; O (N3 N0)
F7 → O (V3 F8 N0) , 0.9 ; O (V3 N0) , 0.1
F8 → O (N3 F9 N0) ; O (N3 N0)
F8 → O (V3 F9 N0) , 0.9 ; O (V3 N0) , 0.1
F9 → O (N3 D0 N0) ; O (N3 N0)
F9 → O (V3 D0 N0) , 0.9 ; O (V3 N0) , 0.1
D0 → O (N3 D1 N0) ; O (N3 N0)
D0 → O (V3 D1 N0) , 0.9 ; O (V3 N0) , 0.1

$$N_0 \rightarrow \begin{matrix} 0 \\ | \\ N_0 \end{matrix} \quad ; \quad 0$$

$$N_1 \rightarrow \begin{matrix} 1 \\ | \\ N_0 \end{matrix} \quad ; \quad 1$$

$$N_2 \rightarrow \begin{matrix} 0 \\ | \\ N_1 \end{matrix}$$

$$N_3 \rightarrow \begin{matrix} 0 \\ | \\ N_2 \end{matrix}$$

$$N_4 \rightarrow \begin{matrix} 0 \\ | \\ N_3 \end{matrix}$$

$$N_5 \rightarrow \begin{matrix} 1 \\ | \\ N_2 \end{matrix}$$

$$N_6 \rightarrow \begin{matrix} 0 \\ | \\ N_5 \end{matrix}$$

$$N_0 \rightarrow \begin{matrix} 0 \\ | \\ N_0 \end{matrix} \quad , 0.75 \quad ; \quad 0 \quad , 0.25$$

$$V_0 \rightarrow \begin{matrix} 0 \\ | \\ N_0 \end{matrix} \quad , 0.6 \quad ; \quad \begin{matrix} 1 \\ | \\ N_0 \end{matrix} \quad , 0.4$$

$$N_1 \rightarrow \begin{matrix} 1 \\ | \\ N_0 \end{matrix} \quad , 0.5 \quad ; \quad \begin{matrix} 0 \\ | \\ N_0 \end{matrix} \quad , 0.2$$

$$1 \quad , 0.2 \quad ; \quad 0 \quad , 0.1$$

$$N_2 \rightarrow \begin{matrix} 0 \\ | \\ N_1 \end{matrix} \quad , 1.0$$

$$N_3 \rightarrow \begin{matrix} 0 \\ | \\ N_2 \end{matrix} \quad , 1.0$$

$$N_4 \rightarrow \begin{matrix} 0 \\ | \\ N_3 \end{matrix} \quad , 1.0$$

$$N_5 \rightarrow \begin{matrix} 1 \\ | \\ N_2 \end{matrix} \quad , 0.7 \quad ; \quad \begin{matrix} 0 \\ | \\ N_2 \end{matrix} \quad , 0.3$$

$$N_6 \rightarrow \begin{matrix} 0 \\ | \\ N_5 \end{matrix} \quad , 1.0$$

$$V_2 \rightarrow \begin{matrix} 1 \\ | \\ N_1 \end{matrix} \quad , 1.0$$

$$V_3 \rightarrow \begin{matrix} 0 \\ | \\ V_3 \end{matrix} \quad , 1.0$$

$G_4' = (V_4', r_4', P_4', X_1)$ over $<\Sigma_4', r_4'>$

$V_4' = \{X_{1,2,\ldots,9,0}, Y_{1,2,\ldots 9,0}\} \cup \Sigma_4'$

$\Sigma_4' = \{A_1, B_0, B_9, C_8, A_7, B_6, C_5, A_4, B_3, C_2, D_1, E_0, E_9, F_8, D_7, E_6, F_5, D_4, E_3, F_2\}$

$r_4' = \{0, 1, 2\}$

P_4':

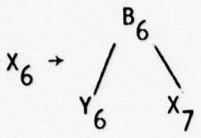
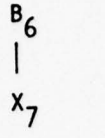
$$X_1 \rightarrow \begin{matrix} & A_1 & \\ / & & \backslash \\ Y_1 & & X_2 \end{matrix} \quad ; \quad \begin{matrix} A_1 \\ | \\ X_2 \end{matrix} \quad ; \quad A_1$$

$$X_2 \rightarrow \begin{matrix} B_0 \\ | \\ X_3 \end{matrix} \quad ; \quad B_0$$

$$X_3 \rightarrow \begin{matrix} B_9 \\ | \\ X_4 \end{matrix} \quad ; \quad B_9$$

$$X_4 \rightarrow \begin{matrix} C_8 \\ | \\ X_5 \end{matrix} \quad ; \quad C_8$$

$$X_5 \rightarrow \begin{matrix} A_7 \\ | \\ X_6 \end{matrix} \quad ; \quad A_7$$

$X_6 \rightarrow$  ;  ; B_6

$X_7 \rightarrow C_5$; C_5

X_8

$X_8 \rightarrow A_4$; A_4

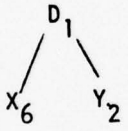
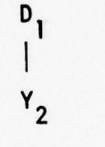
X_9

$X_9 \rightarrow B_3$; B_3

X_0

$X_0 \rightarrow C_2$; C_2

X_1

$Y_1 \rightarrow$  ;  ; D_1

$Y_2 \rightarrow E_0$; E_0

Y_3

$Y_3 \rightarrow E_9$; E_9

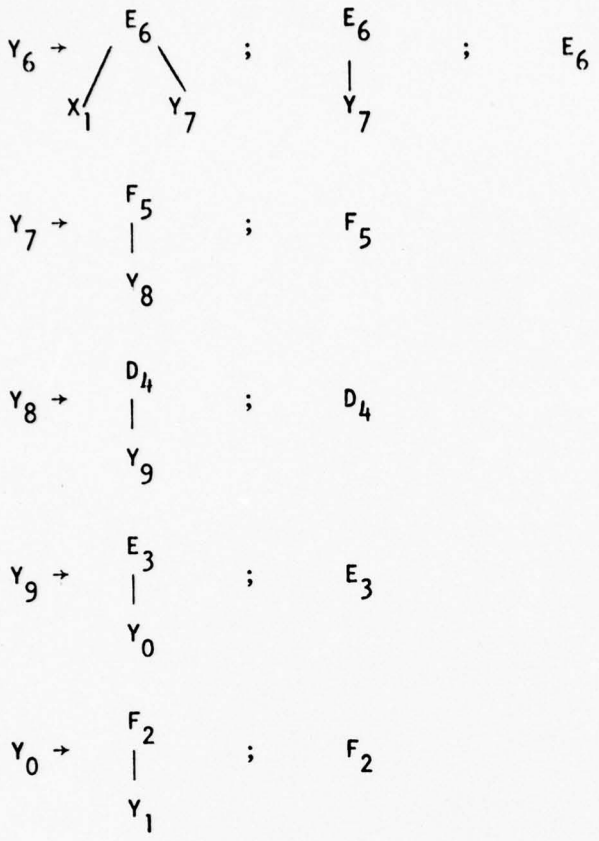
Y_4

$Y_4 \rightarrow F_8$; F_8

Y_5

$Y_5 \rightarrow D_7$; D_7

Y_6



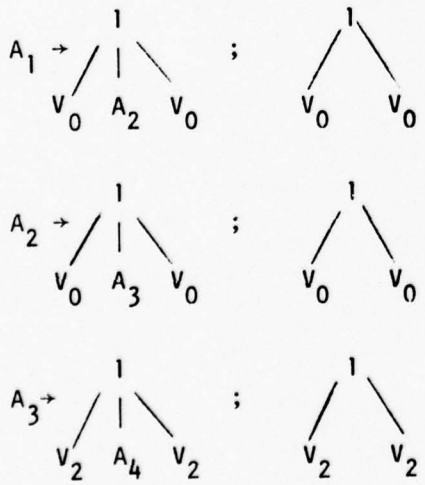
A.2. Grammar for Reptile Skin (pattern D22)

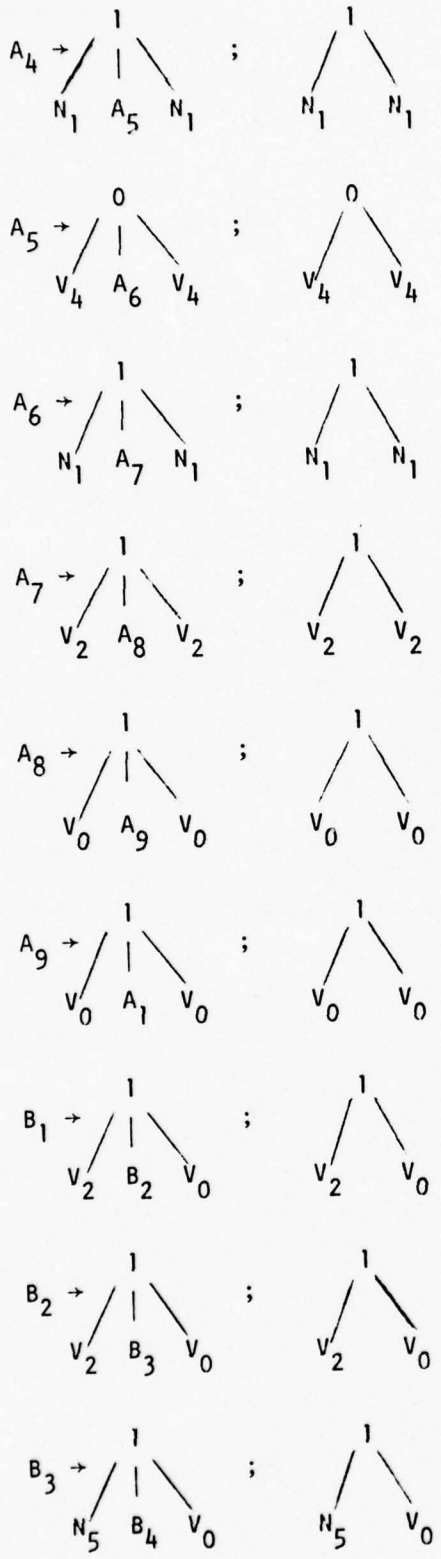
$G_5 = (V_5, r, P_5, S_5)$ over $<\Sigma, r>$

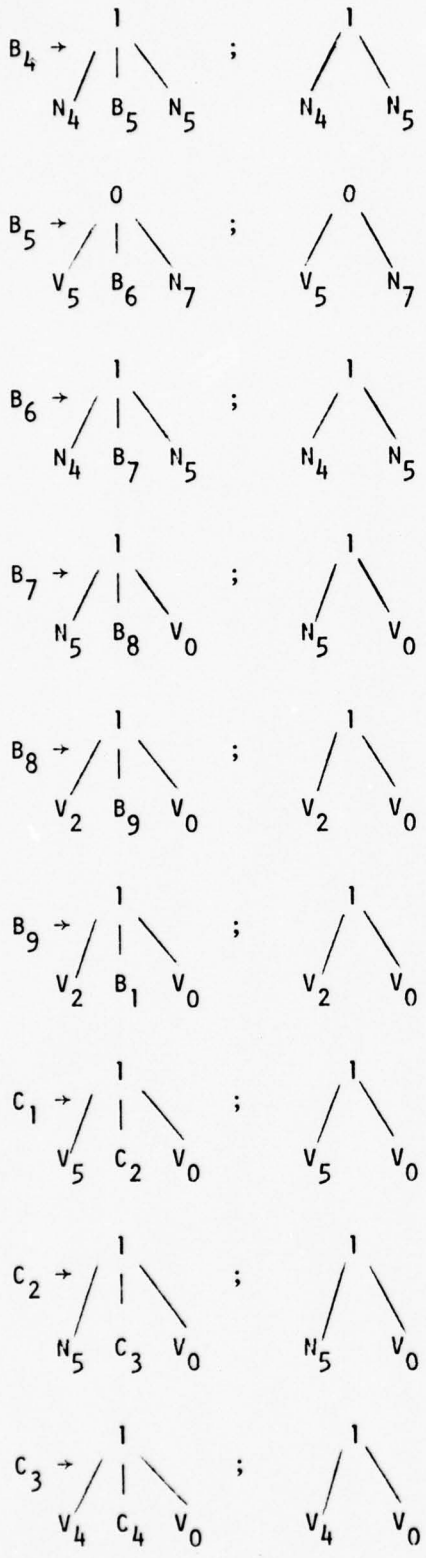
$V_5 = S_5 \cup \Sigma \cup \{N_{0,1,\ldots,9}, {}^{V}0,1,\ldots,5\}$

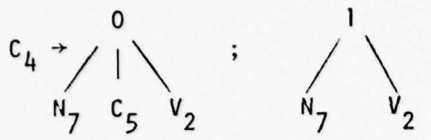
$S_5 = \{X_i \mid X \in \{A, B, C, D, E, F, G, H, I\}, i \in \{1,2,\ldots,9\}\}$

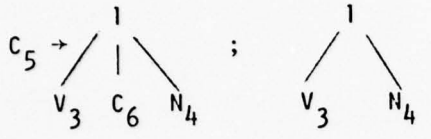
P_5:

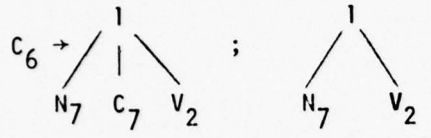


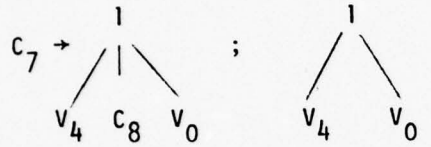

A4 → 1 N1 A5 N1 ; 1 N1 N1
A5 → 0 V4 A6 V4 ; 0 V4 V4
A6 → 1 N1 A7 N1 ; 1 N1 N1
A7 → 1 V2 A8 V2 ; 1 V2 V2
A8 → 1 V0 A9 V0 ; 1 V0 V0
A9 → 1 V0 A1 V0 ; 1 V0 V0
B1 → 1 V2 B2 V0 ; 1 V2 V0
B2 → 1 V2 B3 V0 ; 1 V2 V0
B3 → 1 N5 B4 V0 ; 1 N5 V0

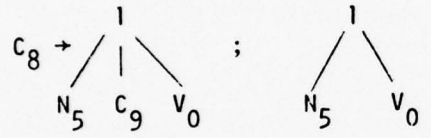

B4 → 1 N4 B5 N5 ; 1 N4 N5
B5 → 0 V5 B6 N7 ; 0 V5 N7
B6 → 1 N4 B7 N5 ; 1 N4 N5
B7 → 1 N5 B8 V0 ; 1 N5 V0
B8 → 1 V2 B9 V0 ; 1 V2 V0
B9 → 1 V2 B1 V0 ; 1 V2 V0
C1 → 1 V5 C2 V0 ; 1 V5 V0
C2 → 1 N5 C3 V0 ; 1 N5 V0
C3 → 1 V4 C4 V0 ; 1 V4 V0

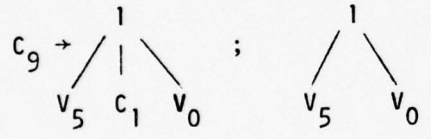

C4 →
0
N7 C5 V2
;
1
N7 V2

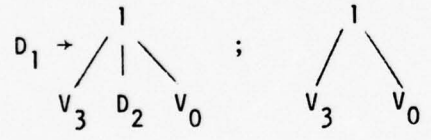

C5 →
1
V3 C6 N4
;
1
V3 N4

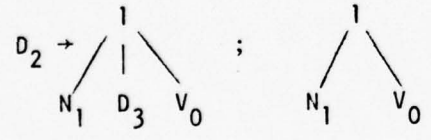

C6 →
1
N7 C7 V2
;
1
N7 V2

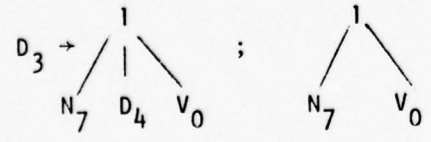

C7 →
1
V4 C8 V0
;
1
V4 V0

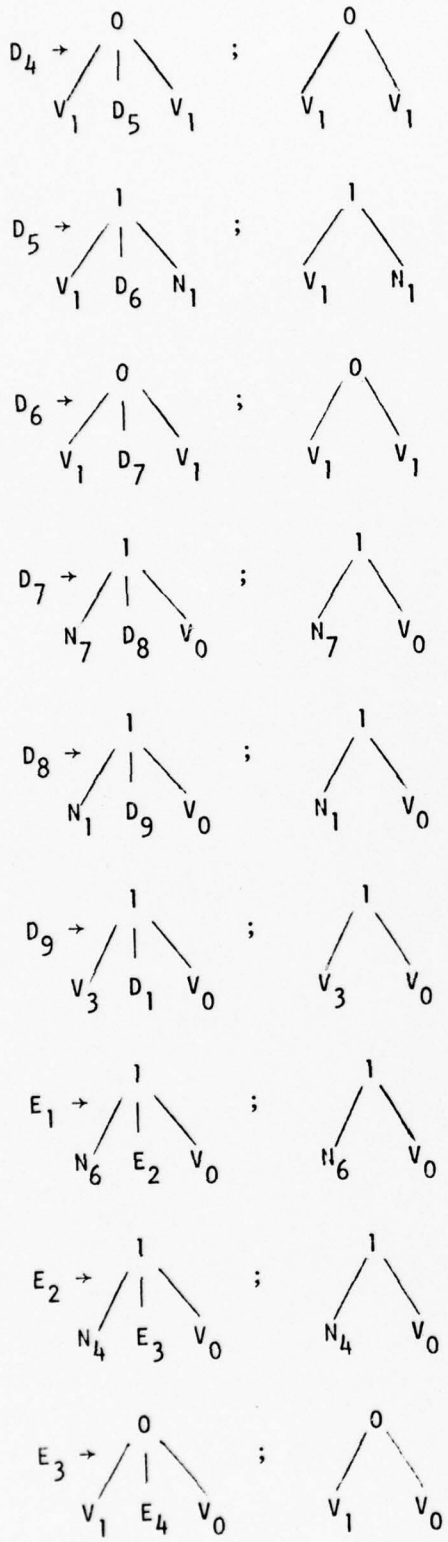

C8 →
1
N5 C9 V0
;
1
N5 V0

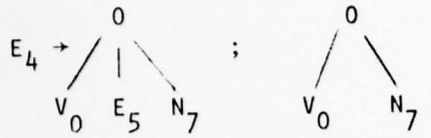

C9 →
1
V5 C1 V0
;
1
V5 V0

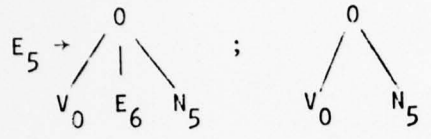

D1 →
1
V3 D2 V0
;
1
V3 V0

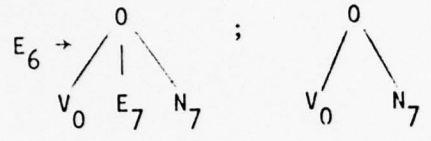

D2 →
1
N1 D3 V0
;
1
N1 V0

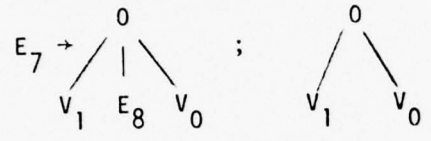

D3 →
1
N7 D4 V0
;
1
N7 V0

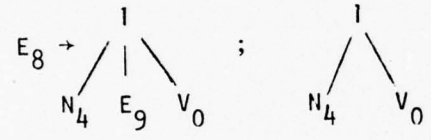

D4 → 0 V1 D5 V1 ; 0 V1 V1
D5 → 1 V1 D6 N1 ; 1 V1 N1
D6 → 0 V1 D7 V1 ; 0 V1 V1
D7 → 1 N7 D8 V0 ; 1 N7 V0
D8 → 1 N1 D9 V0 ; 1 N1 V0
D9 → 1 V3 D1 V0 ; 1 V3 V0
E1 → 1 N6 E2 V0 ; 1 N6 V0
E2 → 1 N4 E3 V0 ; 1 N4 V0
E3 → 0 V1 E4 V0 ; 0 V1 V0

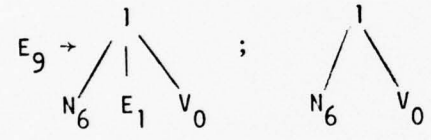

E4 →
0
V0 E5 N7
;
0
V0 N7

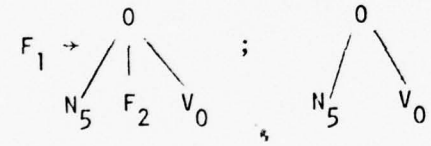

E5 →
0
V0 E6 N5
;
0
V0 N5

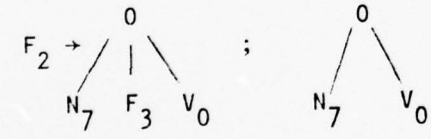

E6 →
0
V0 E7 N7
;
0
V0 N7

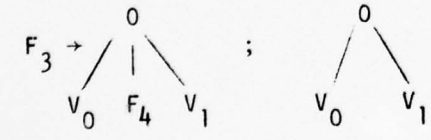

E7 →
0
V1 E8 V0
;
0
V1 V0

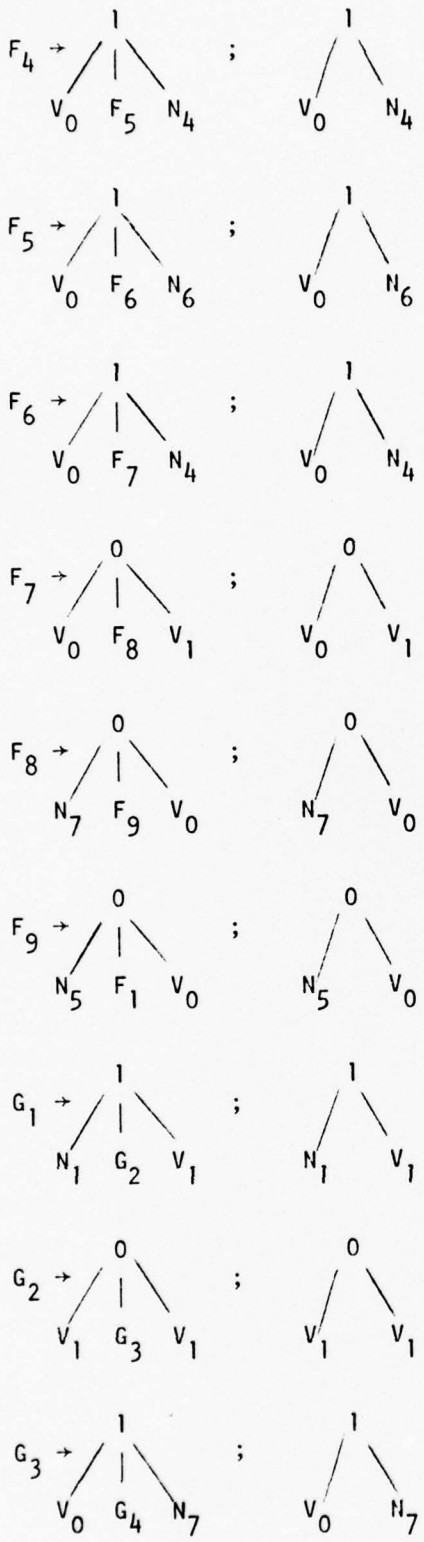

E8 →
1
N4 E9 V0
;
1
N4 V0

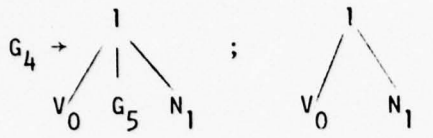

E9 →
1
N6 E1 V0
;
1
N6 V0

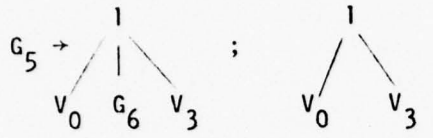

F1 →
0
N5 F2 V0
;
0
N5 V0

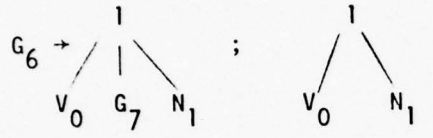

F2 →
0
N7 F3 V0
;
0
N7 V0

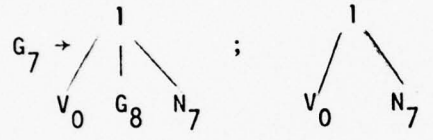

F3 →
0
V0 F4 V1
;
0
V0 V1

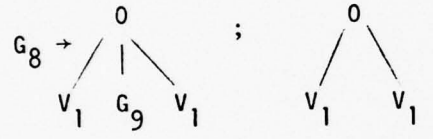

F4 → 1 V0 F5 N4 ; 1 V0 N4
F5 → 1 V0 F6 N6 ; 1 V0 N6
F6 → 1 V0 F7 N4 ; 1 V0 N4
F7 → 0 V0 F8 V1 ; 0 V0 V1
F8 → 0 N7 F9 V0 ; 0 N7 V0
F9 → 0 N5 F1 V0 ; 0 N5 V0
G1 → 1 N1 G2 V1 ; 1 N1 V1
G2 → 0 V1 G3 V1 ; 0 V1 V1
G3 → 1 V0 G4 N7 ; 1 V0 N7

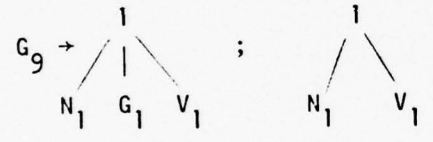

G4 → 1 V0 G5 N1 ; 1 V0 N1

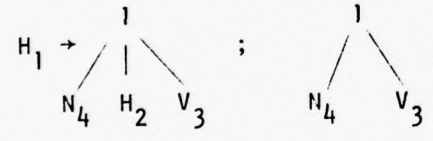

G5 → 1 V0 G6 V3 ; 1 V0 V3

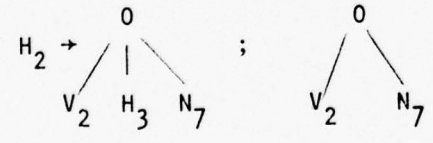

G6 → 1 V0 G7 N1 ; 1 V0 N1

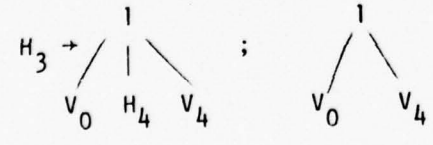

G7 → 1 V0 G8 N7 ; 1 V0 N7

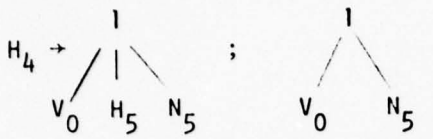

G8 → 0 V1 G9 V1 ; 0 V1 V1

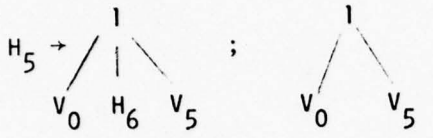

G9 → 1 N1 G1 V1 ; 1 N1 V1

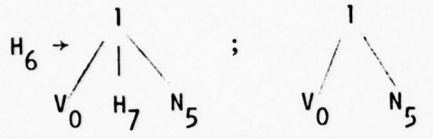

H1 → 1 N4 H2 V3 ; 1 N4 V3


H2 → 0 V2 H3 N7 ; 0 V2 N7

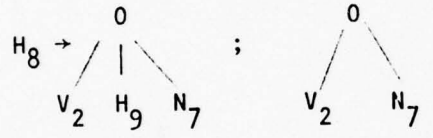

H3 → 1 V0 H4 V4 ; 1 V0 V4

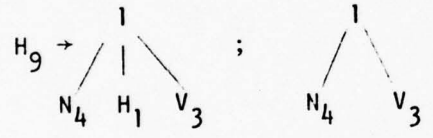

H4 → 1 V0 H5 N5 ; 1 V0 N5

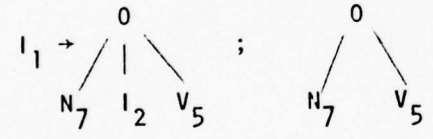

H5 → 1 V0 H6 V5 ; 1 V0 V5

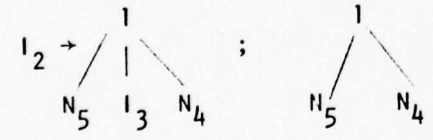

H6 → 1 V0 H7 N5 ; 1 V0 N5

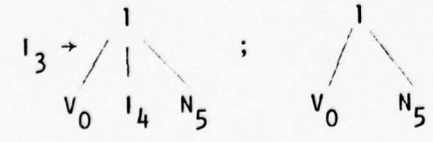

H7 → 1 V0 H8 V4 ; 1 V0 V4

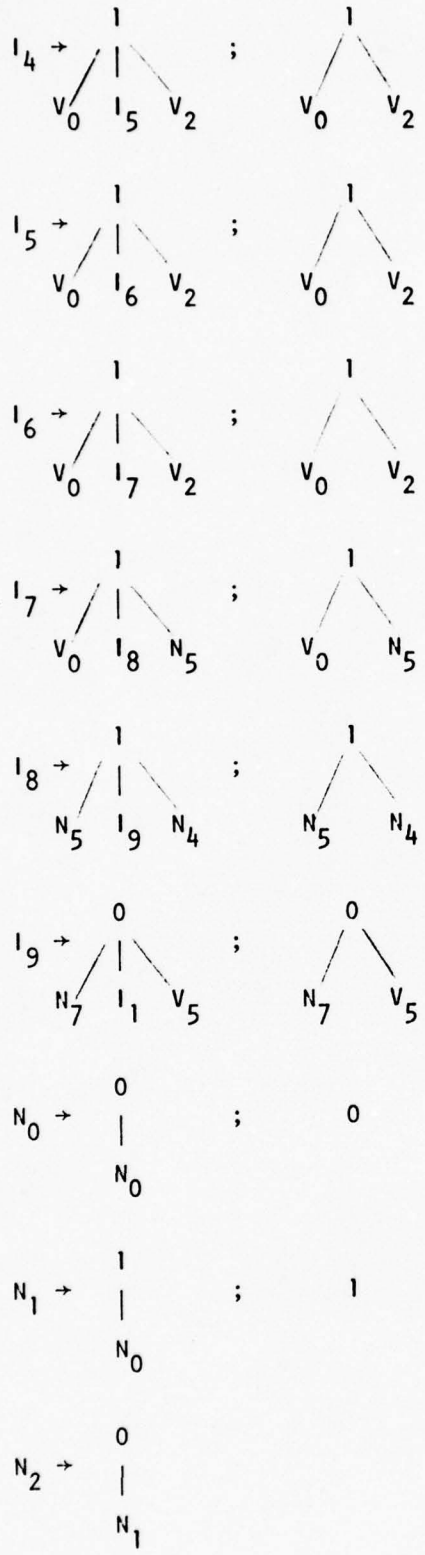

H8 → 0 V2 H9 N7 ; 0 V2 N7


H9 → 1 N4 H1 V3 ; 1 N4 V3


I1 → 0 N7 I2 V5 ; 0 N7 V5


I2 → 1 N5 I3 N4 ; 1 N5 N4


I3 → 1 V0 I4 N5 ; 1 V0 N5


I4 → 1 V0 I5 V2 ; 1 V0 V2
I5 → 1 V0 I6 V2 ; 1 V0 V2
I6 → 1 V0 I7 V2 ; 1 V0 V2
I7 → 1 V0 I8 N5 ; 1 V0 N5
I8 → 1 N5 I9 N4 ; 1 N5 N4
I9 → 0 N7 I1 V5 ; 0 N7 V5
N0 → 0 N0 ; 0
N1 → 1 N0 ; 1
N2 → 0 N1

$N_3 \rightarrow \begin{matrix} 0 \\ | \\ N_2 \end{matrix}$

$N_4 \rightarrow \begin{matrix} 0 \\ | \\ N_3 \end{matrix}$

$N_5 \rightarrow \begin{matrix} 1 \\ | \\ N_1 \end{matrix}$

$N_0 \rightarrow \begin{matrix} 0 \\ | \\ N_5 \end{matrix}$

$N_7 \rightarrow \begin{matrix} 0 \\ | \\ N_6 \end{matrix}$

$N_8 \rightarrow \begin{matrix} 1 \\ | \\ N_2 \end{matrix}$

$N_9 \rightarrow \begin{matrix} 0 \\ | \\ N_8 \end{matrix}$

$V_0 \rightarrow \begin{matrix} 1 \\ | \\ V_0 \end{matrix} \quad ; \quad 1$

$V_1 \rightarrow \begin{matrix} 0 \\ | \\ V_0 \end{matrix}$

$V_2 \rightarrow \begin{matrix} 1 \\ | \\ N_5 \end{matrix}$

$V_3 \rightarrow \begin{matrix} 1 \\ | \\ N_6 \end{matrix}$

$V_4 \rightarrow \begin{matrix} 1 \\ | \\ N_3 \end{matrix}$

$V_5 \rightarrow \begin{matrix} 1 \\ | \\ N_8 \end{matrix}$

Appendix B. Discrimination Grammars for Pattern D22, D34, D38, and D68

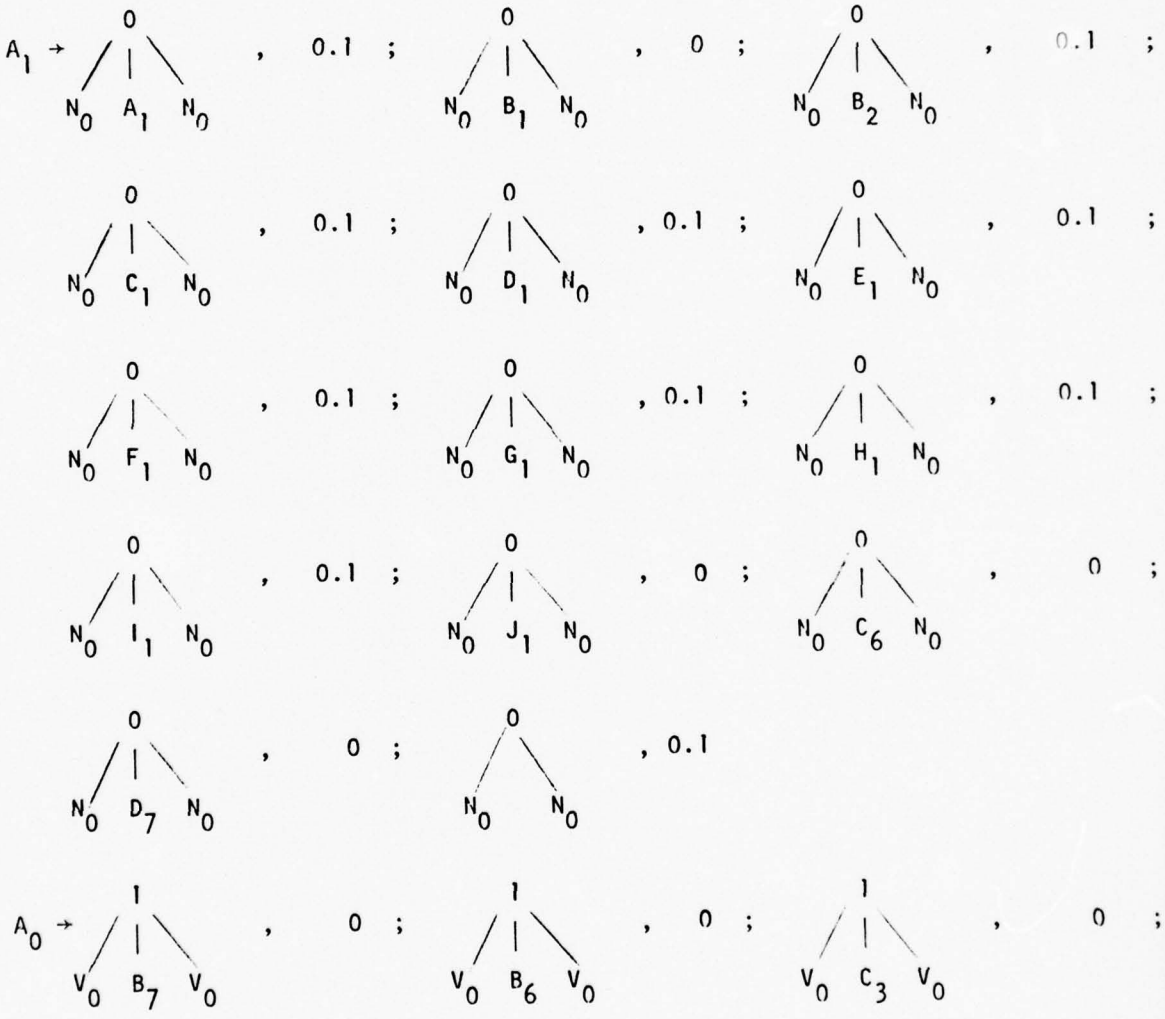
B-1. Grammar for the Synthesis and Discrimination of Water Pattern (D38)

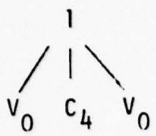
$G_6 = (V_6, r, P_6, S_6)$ over $<\Sigma, r>$

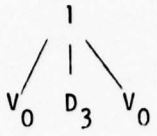
$V_6 = S_6 \cup \{N_{0,1,\ldots,9}, V_{0,1,\ldots,5}\} \cup \Sigma$

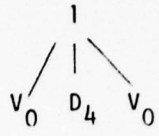
$S_6 = \{A_{0,1}, B_{1,2,\ldots,7}, C_{1,2,\ldots,6}, D_{1,2,\ldots,7}, E_{1,2,\ldots,5}, F_{1,2,\ldots,5}, G_{1,2,3}, H_{1,2,3}, I_1, J_1\}$

P_6:

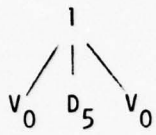




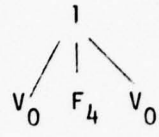
, 0 ;


, 0 ;


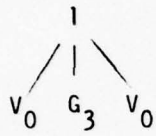
, 0 ;

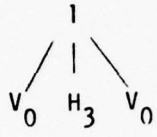


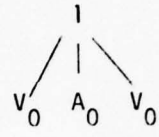
, 0 ;


, 0 ;


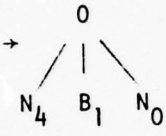
, 0 ;

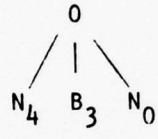


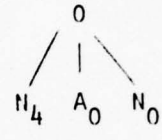
, 0 ;


, 0 ;


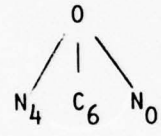
, 0

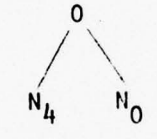
$B_1 \rightarrow$ 

, 0 ;


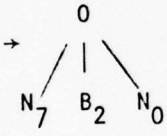
, 0 ;


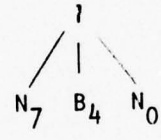
, 0 ;

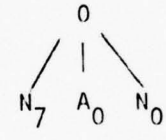


, 0 ;


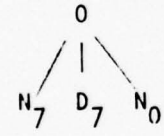
, 0

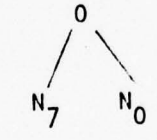
$B_2 \rightarrow$ 

, 0.4 ;


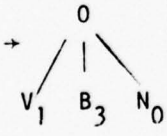
, 0.3 ;


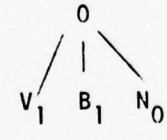
, 0.2 ;

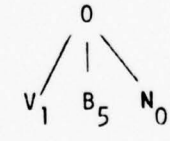


, 0 ;


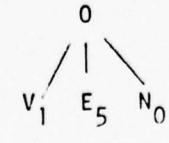
, 0.1 ;

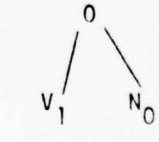
$B_3 \rightarrow$ 

, 0 ;


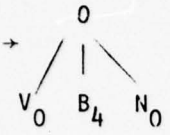
, 0 ;


, 0 ;

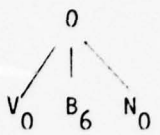
, 0 ;


, 0 ;


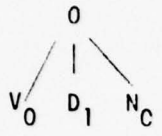
, 0

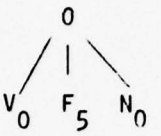
$B_4 \rightarrow$ 

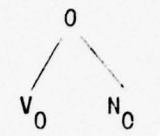
, 0.4 ; 

, 0.2 ; 

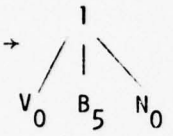
, 0.15 ;

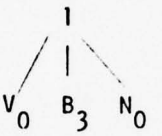


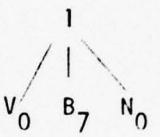
, 0.15 ; 

, 0 ; 

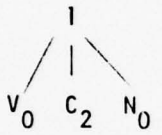
, 0.1

$B_5 \rightarrow$ 

, 0 ; 

, 0 ; 

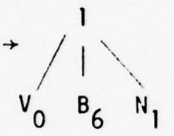
, 0 ;

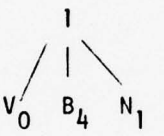


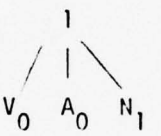
, 0 ; 

, 0 ; 

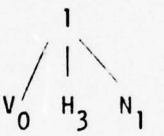
, 0

$B_6 \rightarrow$ 

, 0.4 ; 

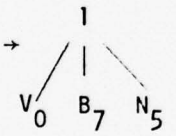
, 0.3 ; 

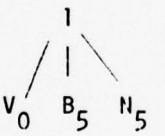
, 0 ;

, 0.2 ; 

, 0 ; 

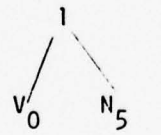
, 0.1

$B_7 \rightarrow$ 

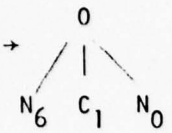
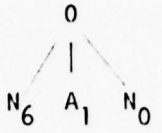
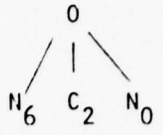
, 0 ; 

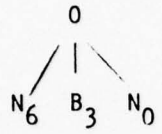
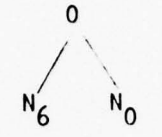
, 0 ; 

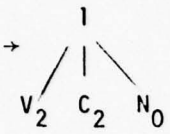
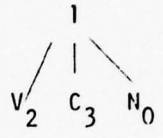
, 0 ;

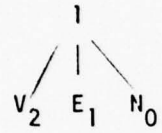
, 0 ; 

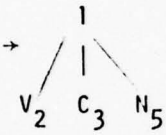
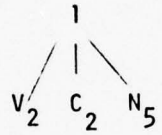
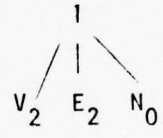
, 0

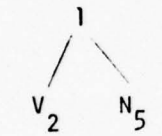
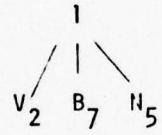
$C_1 \rightarrow$  , 0.4 ;  , 0.2 ;  , 0.3 ;

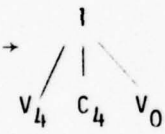
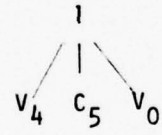
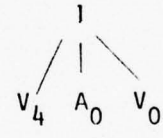
 , 0 ;  , 0.1

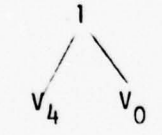
$C_2 \rightarrow$  , 0.4 ;  , 0.2 ;  , 0.15 ;

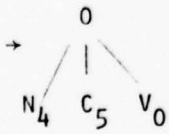
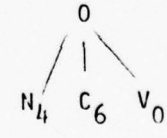
 , 0.15 ;  , 0.1

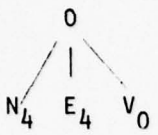
$C_3 \rightarrow$  , 0.4 ;  , 0.3 ;  , 0.3 ;

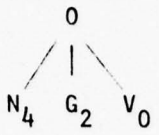
 , 0.1 ;  , 0

$C_4 \rightarrow$  , 0 ;  , 0 ;  , 0 ;

 , 0 ;  , 0

$C_5 \rightarrow$  , 0 ;  , 0 ;  , 0 ;

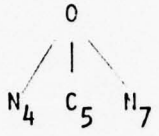


, 0 ;


, 0 ;

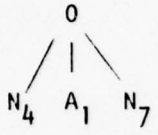

, 0

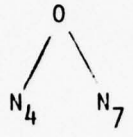
$C_6 \rightarrow$


, 0 ;


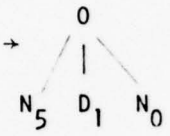
, 0 ;

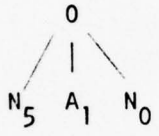

, 0 ;

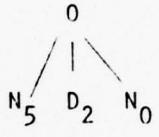


, 0 ;


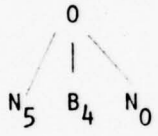
, 0

$D_1 \rightarrow$


, 0.4 ;


, 0.2 ;


, 0.15 ;

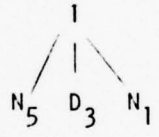


, 0.15 ;

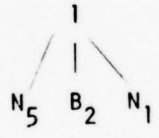

, 0.1

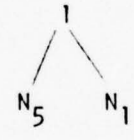
$D_2 \rightarrow$


, 0.4 ;


, 0.1 ;


, 0.2 ;

, 0.1 ;


, 0.1 ;


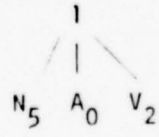
, 0.1

$D_3 \rightarrow$


, 0.4 ;

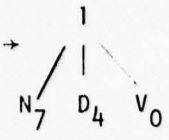
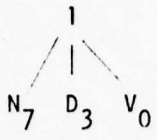
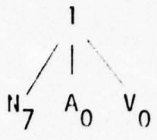

, 0.15 ;

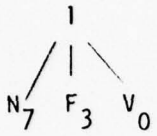

, 0 ;

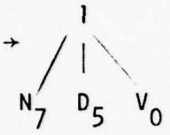
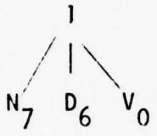
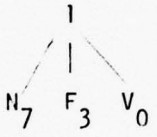
, 0.15 ;


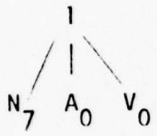
, 0.2 ;

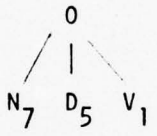
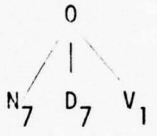

, 0.1

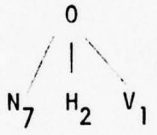
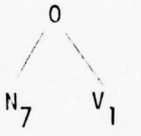
$D_4 \rightarrow$  , 0 ;  , 0 ;  , 0 ;

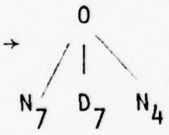
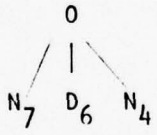
 , 0 ;  , 0

$D_5 \rightarrow$  , 0 ;  , 0 ;  , 0 ;

 , 0 ;  , 0

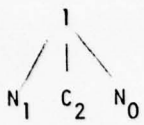
$D_6 \rightarrow$  , 0 ;  , 0 ;  , 0 ;

 , 0 ;  , 0 ;  , 0

$D_7 \rightarrow$  , 0 ;  , 0 ;  , 0 ;

 , 0 ;  , 0

$E_1 \rightarrow$  , 0.4 ;  ,0.15 ;  , 0.2 ;



, 0.15 ;

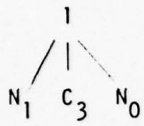
, 0.1

$E_2 \rightarrow$

, 0.4 ;

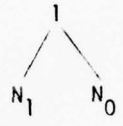
,0.15 ;

, 0.1



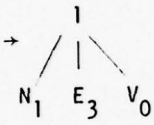
, 0.15 ;

, 0.1 ;

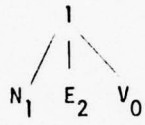


, 0.1

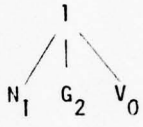
$E_3 \rightarrow$



, 0.4 ;



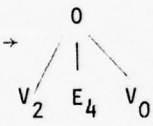
, 0.3 ;



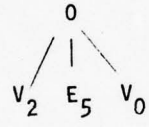
, 0.2 ;

, 0.1

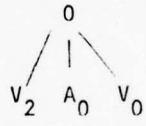
$E_4 \rightarrow$



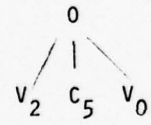
, 0 ;



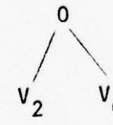
, 0 ;



, 0 ;

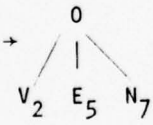


, 0 ;



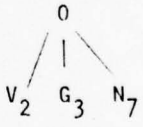
, 0

$E_5 \rightarrow$



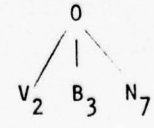
, 0 ;

, 0 ;

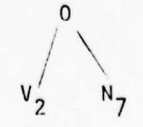


, 0 ;

, 0 ;

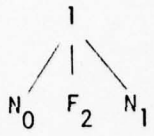


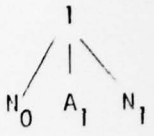
, 0 ;



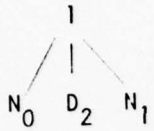
, 0

$F_1 \rightarrow$ 

, 0.4 ; 

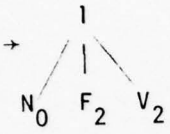
,0.15 ; 

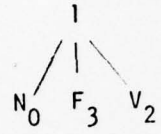
, 0.2 ;



, 0.15 ; 

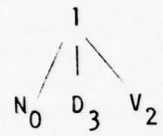
, 0.1

$F_2 \rightarrow$ 

, 0.4 ; 

,0.15 ; 

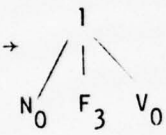
, 0.1 ;



, 0.15 ; 

, 0.1 ; 

, 0.1

$F_3 \rightarrow$ 

, 0.4 ; 

, 0.3 ; 

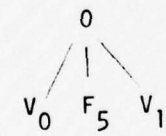
, 0 ;

, 0.2 ; 

, 0 ; 

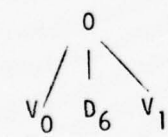
, 0.1

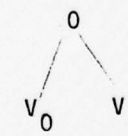
$F_4 \rightarrow$ 

, 0 ; 

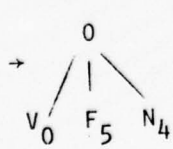
, 0 ; 

, 0 ;



, 0 ; 

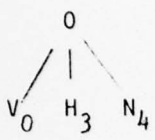
, 0

$F_5 \rightarrow$ 

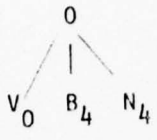
, 0 ; 

, 0 ; 

, 0 ;



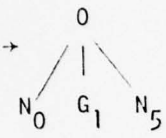
, 0 ;



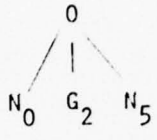
, 0 ;

, 0

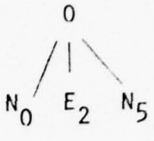
$G_1 \rightarrow$



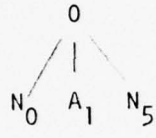
, 0.4 ;



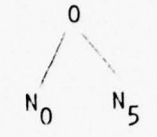
,0.15 ;



, 0.15 ;

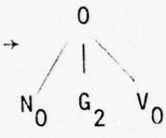


, 0.2 ;

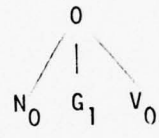


, 0.1

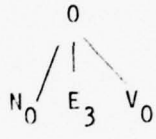
$G_2 \rightarrow$



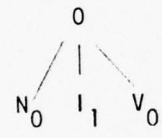
, 0.4 ;



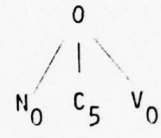
, 0.1 ;



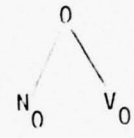
, 0.3 ;



, 1 ;

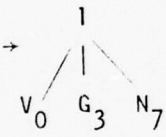


, 0 ;



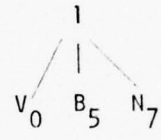
, 0.1

$G_3 \rightarrow$



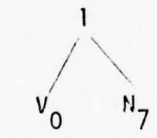
, 0 ;

, 0 ;



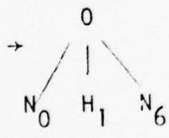
, 0 ;

, 0 ;

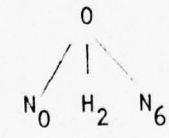


, 0

$H_1 \rightarrow$

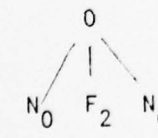


, 0.4 ;

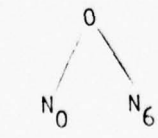


,0.15 ;

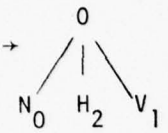
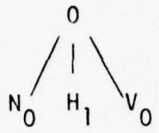
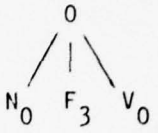
, 0.2 ;

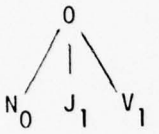
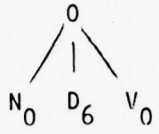
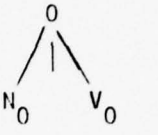


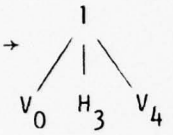
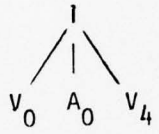
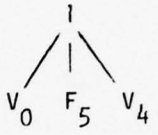
, 0.15 ;

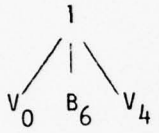


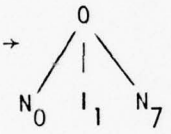
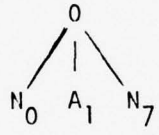
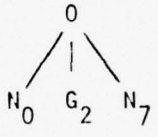
, 0.1

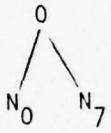
$H_2 \rightarrow$  , 0.4 ;  , 0.1 ;  , 0.3 ;

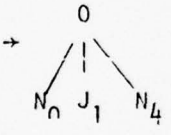
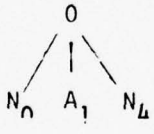
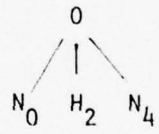
 , 0.1 ;  , 0 ;  , 0.1

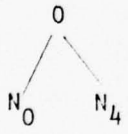
$H_3 \rightarrow$  , 0 ;  , 0 ;  , 0 ;

 , 0 ;  , 0

$I_1 \rightarrow$  , 0.4 ;  , 0.2 ;  , 0.3 ;

 , 0.1

$J_1 \rightarrow$  , 0.4 ;  , 0.2 ;  , 0.3 ;

 , 0.1

Production rules with left-hand side nonterminals as $N_{0,1,\ldots,9}$ or $V_{0,1,\ldots,5}$ are the same as those in grammar G_5 in Appendix A-2.

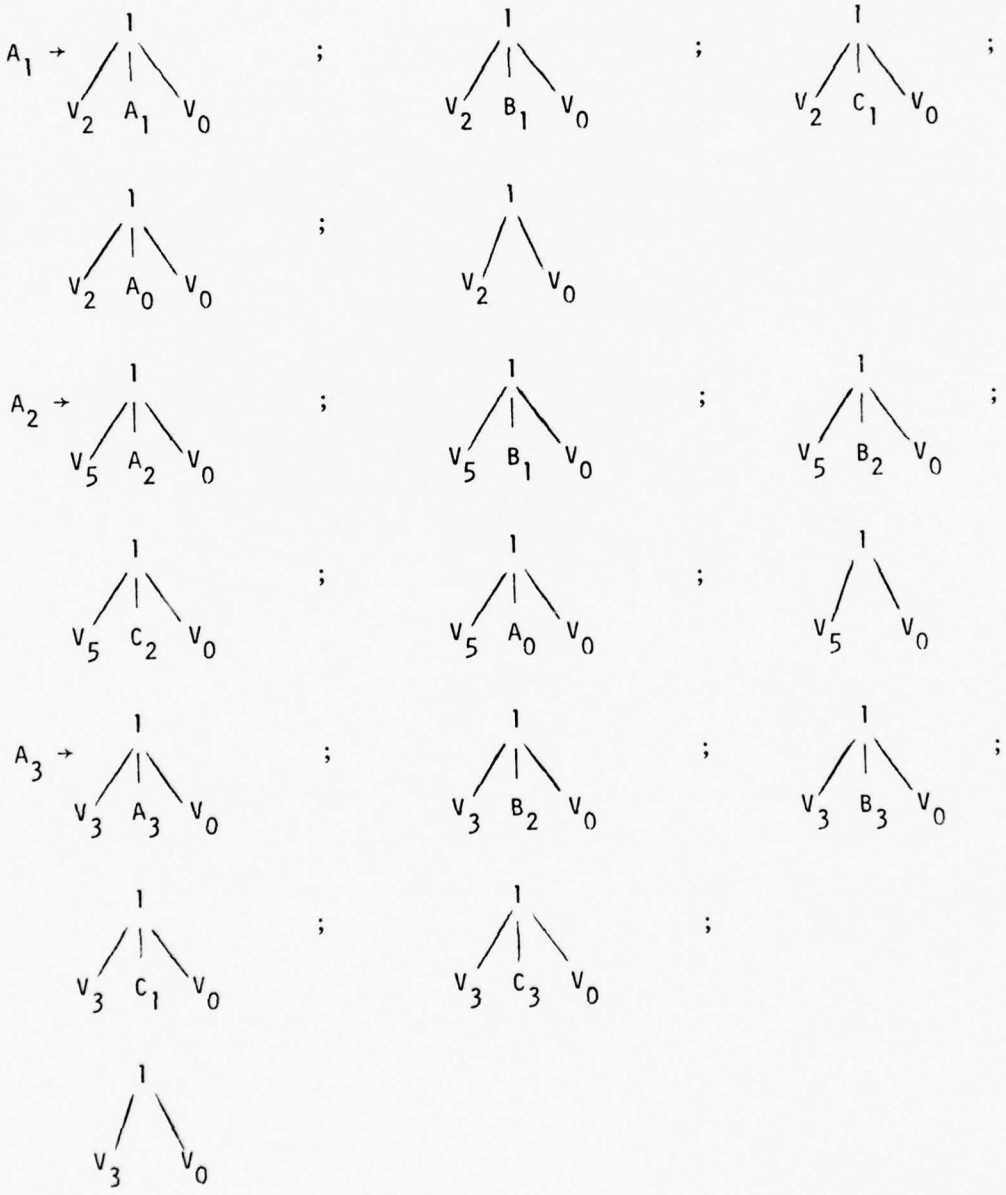
B-2. Discrimination Grammar for Pattern D22

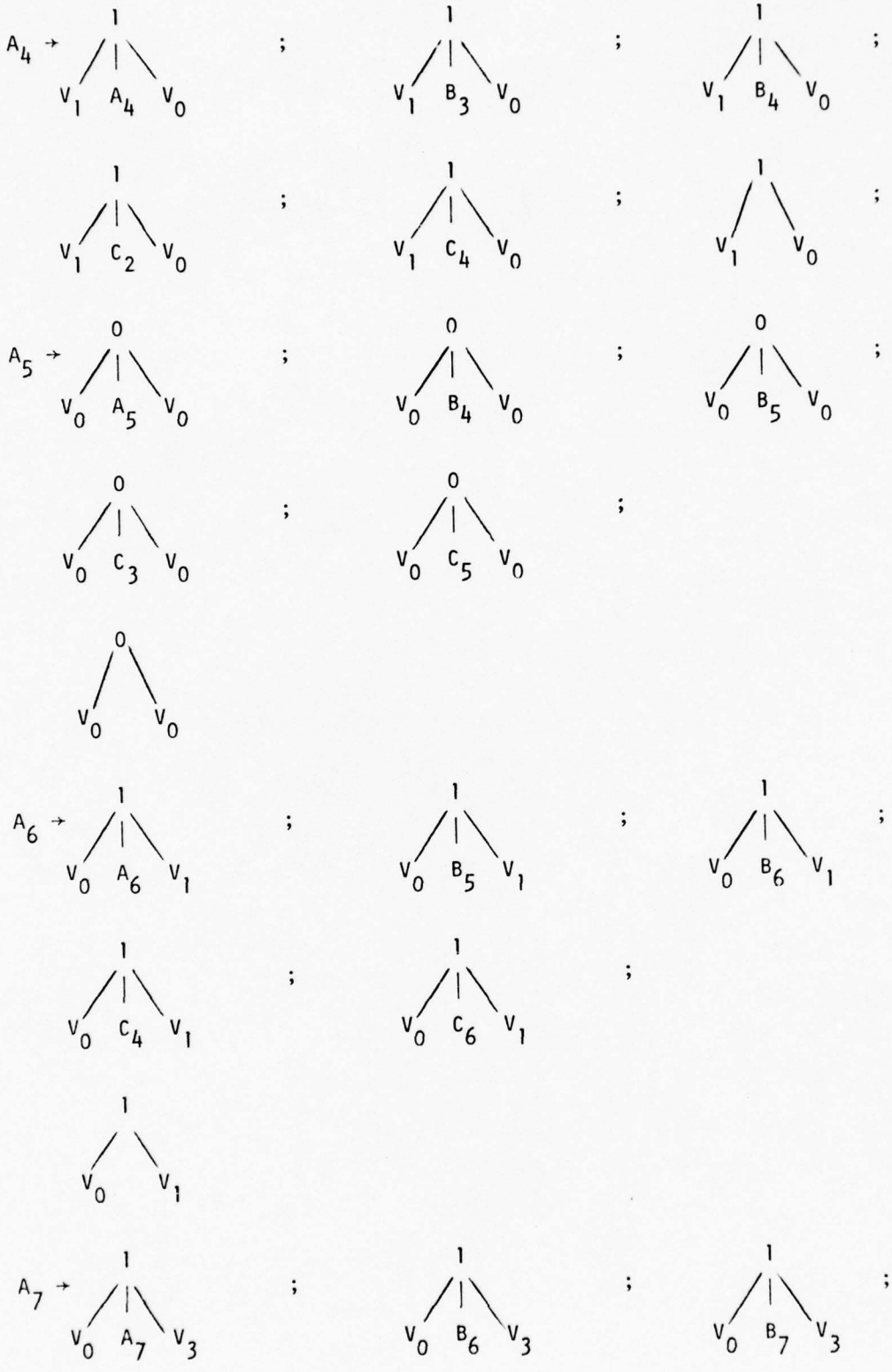
$G_{22} = (V_{22}, r, P_{22}, S_{22})$ over $<\Sigma, r>$

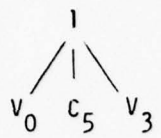
$V_{22} = S_{22} \cup \{N_{0,1,\ldots,9}, V_{0,1,\ldots,5}\} \cup \Sigma$

$S_{22} = A_{0,1,\ldots,9}, B_{1,2,\ldots,8}, C_{1,2,\ldots,7}, D_{1,2,\ldots,6}, E_{1,2,\ldots,6},$
$F_{1,2,\ldots,6}, G_{1,2,\ldots,6}, H_{1,2,\ldots,6}, I_{1,2,3}.$

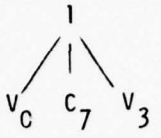
P_{22}:




A4 → 1 V1 A4 V0 ; 1 V1 B3 V0 ; 1 V1 B4 V0 ;
1 V1 C2 V0 ; 1 V1 C4 V0 ; 1 V1 V0 ;
A5 → 0 V0 A5 V0 ; 0 V0 B4 V0 ; 0 V0 B5 V0 ;
0 V0 C3 V0 ; 0 V0 C5 V0 ;
0 V0 V0
A6 → 1 V0 A6 V1 ; 1 V0 B5 V1 ; 1 V0 B6 V1 ;
1 V0 C4 V1 ; 1 V0 C6 V1 ;
1 V0 V1
A7 → 1 V0 A7 V3 ; 1 V0 B6 V3 ; 1 V0 B7 V3 ;

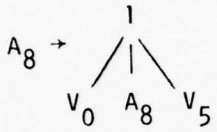

1
V0 C5 V3

;

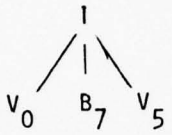

1
V0 C7 V3

;

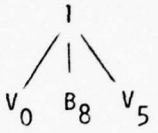

1
V0 V3


A8 →
1
V0 A8 V5

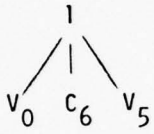
;


1
V0 B7 V5

;

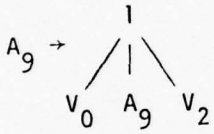

1
V0 B8 V5

;

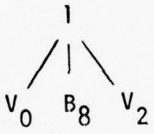

1
V0 C6 V5

;

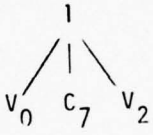

1
V0 V5


A9 →
1
V0 A9 V2

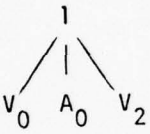
;


1
V0 B8 V2

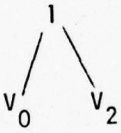
;

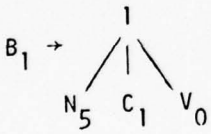

1
V0 C7 V2

;

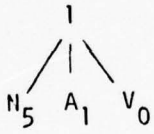

1
V0 A0 V2

;

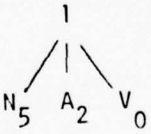

1
V0 V2


B1 →
1
N5 C1 V0

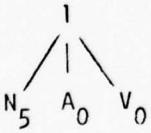
;


1
N5 A1 V0

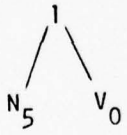
;

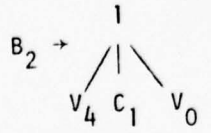

1
N5 A2 V0

;

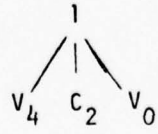

1
N5 A0 V0

;

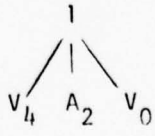

1
N5 V0


B2 →
1
V4 C1 V0

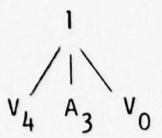
;


1
V4 C2 V0

;

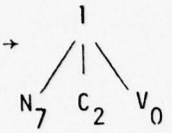

1
V4 A2 V0

;

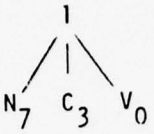


;

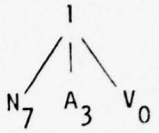
$B_3 \rightarrow$



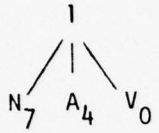
;



;

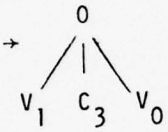


;

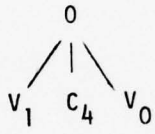


;

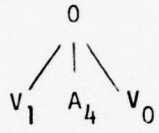
$B_4 \rightarrow$



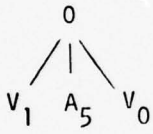
;



;

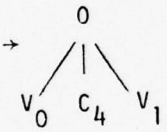


;

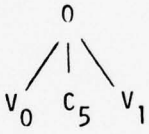


;

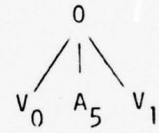
$B_5 \rightarrow$



;



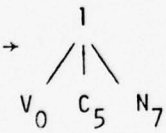
;



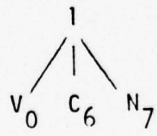
;

;

$B_6 \rightarrow$



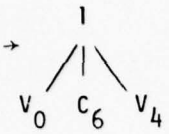
;

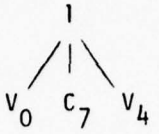


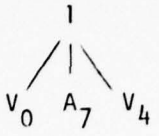
;

;

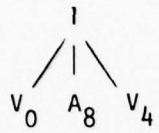
;

$B_7 \rightarrow$ 

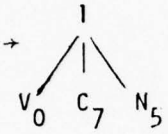
;


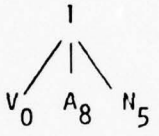
;


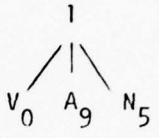
;



;


$B_8 \rightarrow$ 

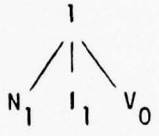
;


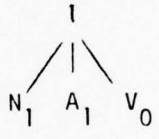
;


;

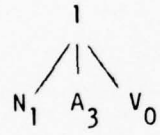
;

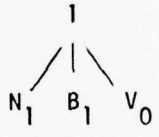

$C_1 \rightarrow$ 

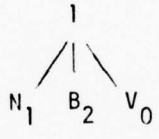
;


;


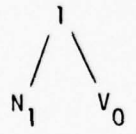
;

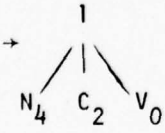


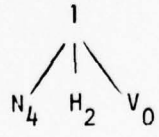
;


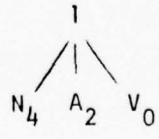
;


;

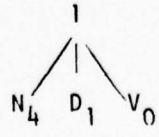


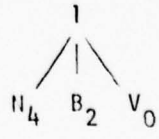
$C_2 \rightarrow$ 

;


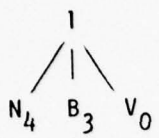
;


;

;


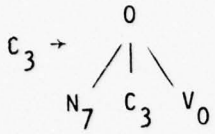
;


;

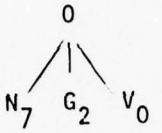

1
N4 B3 V0

;

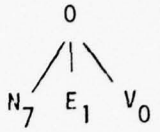

1
N4 V0


C3 →
0
N7 C3 V0

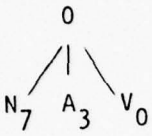
;


0
N7 G2 V0

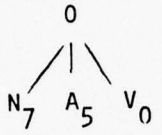
;


0
N7 E1 V0

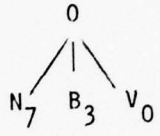
;


0
N7 A3 V0

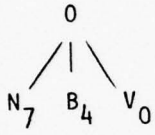
;


0
N7 A5 V0

;

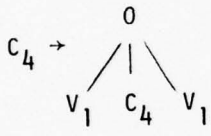

0
N7 B3 V0

;

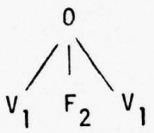

0
N7 B4 V0

;

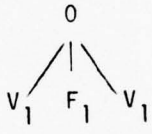

0
N7 V0


C4 →
0
V1 C4 V1

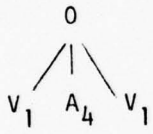
;


0
V1 F2 V1

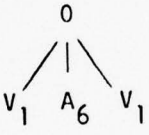
;


0
V1 F1 V1

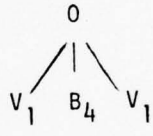
;


0
V1 A4 V1

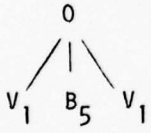
;


0
V1 A6 V1

;

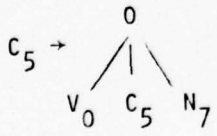

0
V1 B4 V1

;

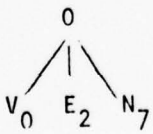

0
V1 B5 V1

;

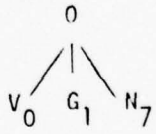

0
V1 V1


C5 →
0
V0 C5 N7

;


0
V0 E2 N7

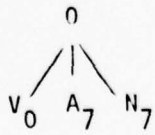
;


0
V0 G1 N7

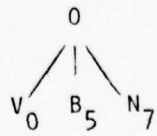
;


0
V0 A5 N7

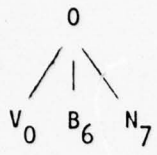
;


0
V0 A7 N7

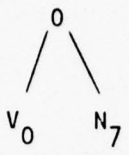
;


0
V0 B5 N7

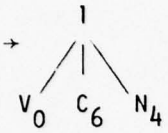
;



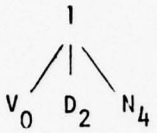
;



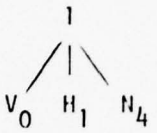
$C_6 \rightarrow$



;

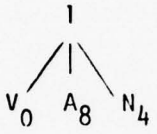


;

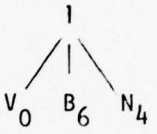


;

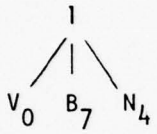
;



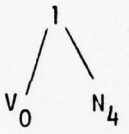
;



;



;

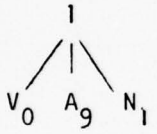


$C_7 \rightarrow$

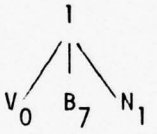
;

;

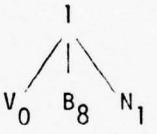
;



;

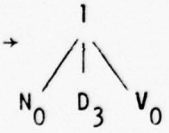


;

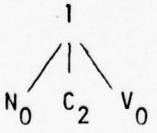


;

$D_1 \rightarrow$

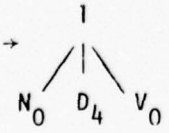


;

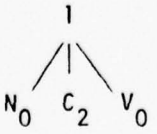


;

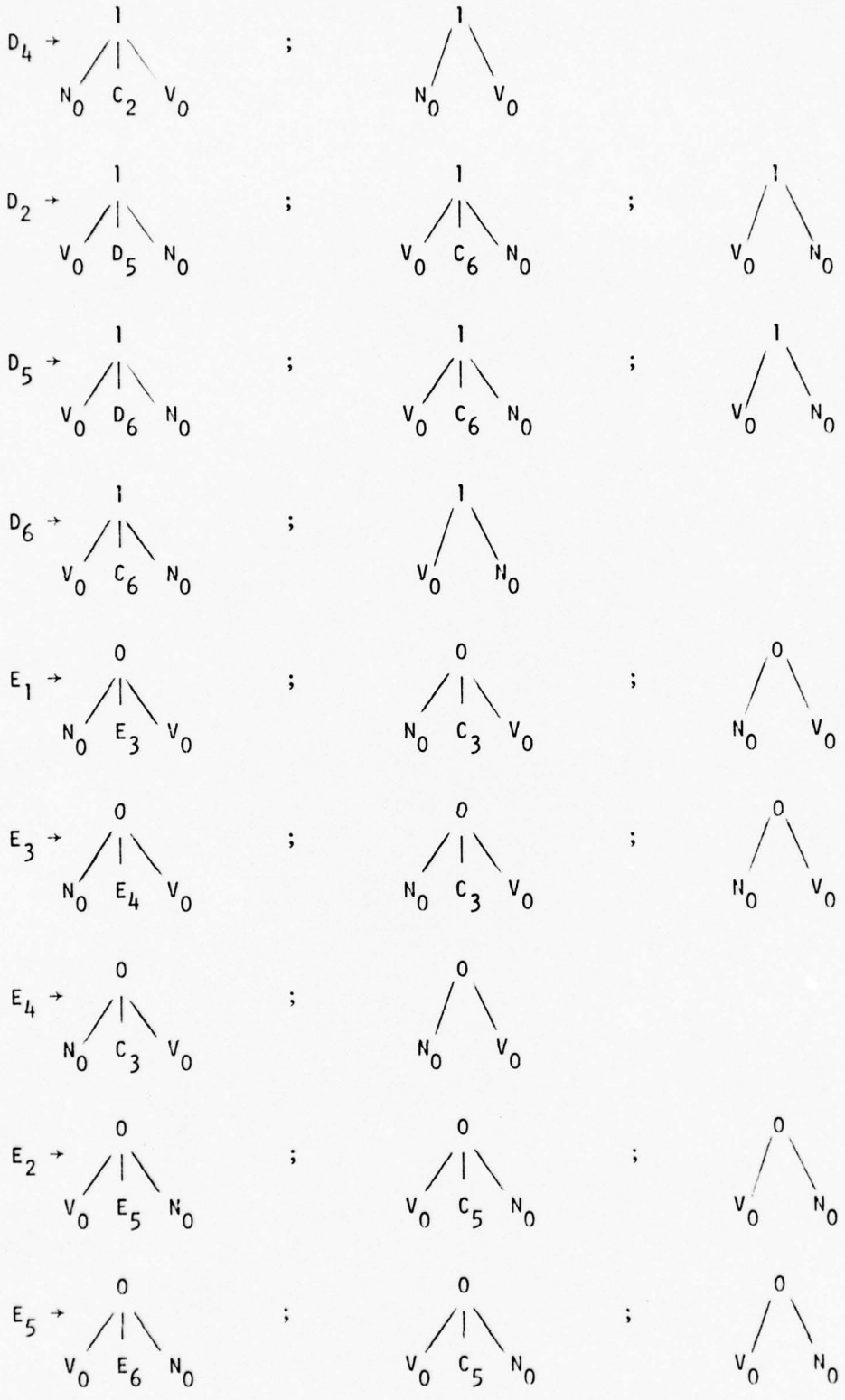
$D_3 \rightarrow$

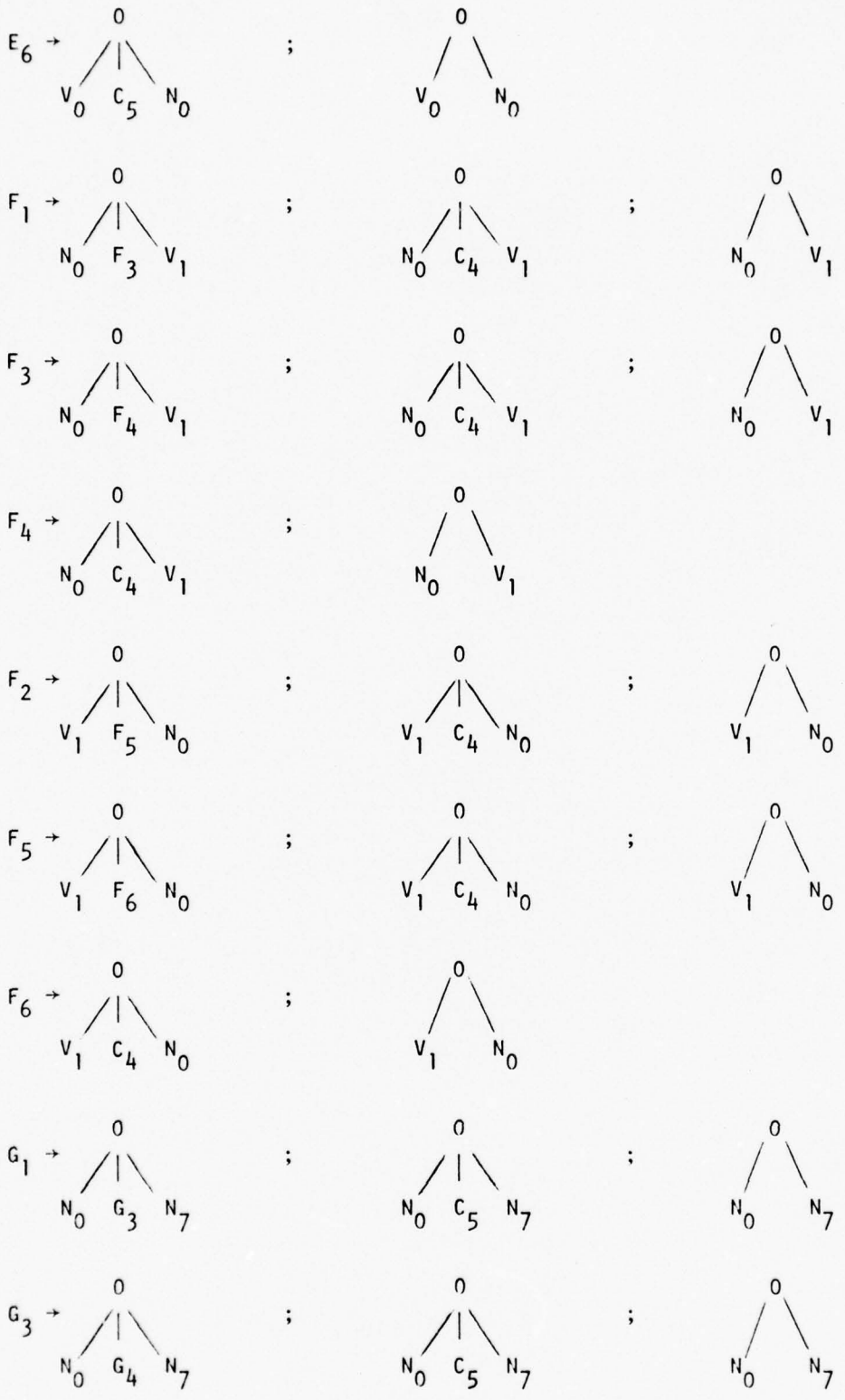


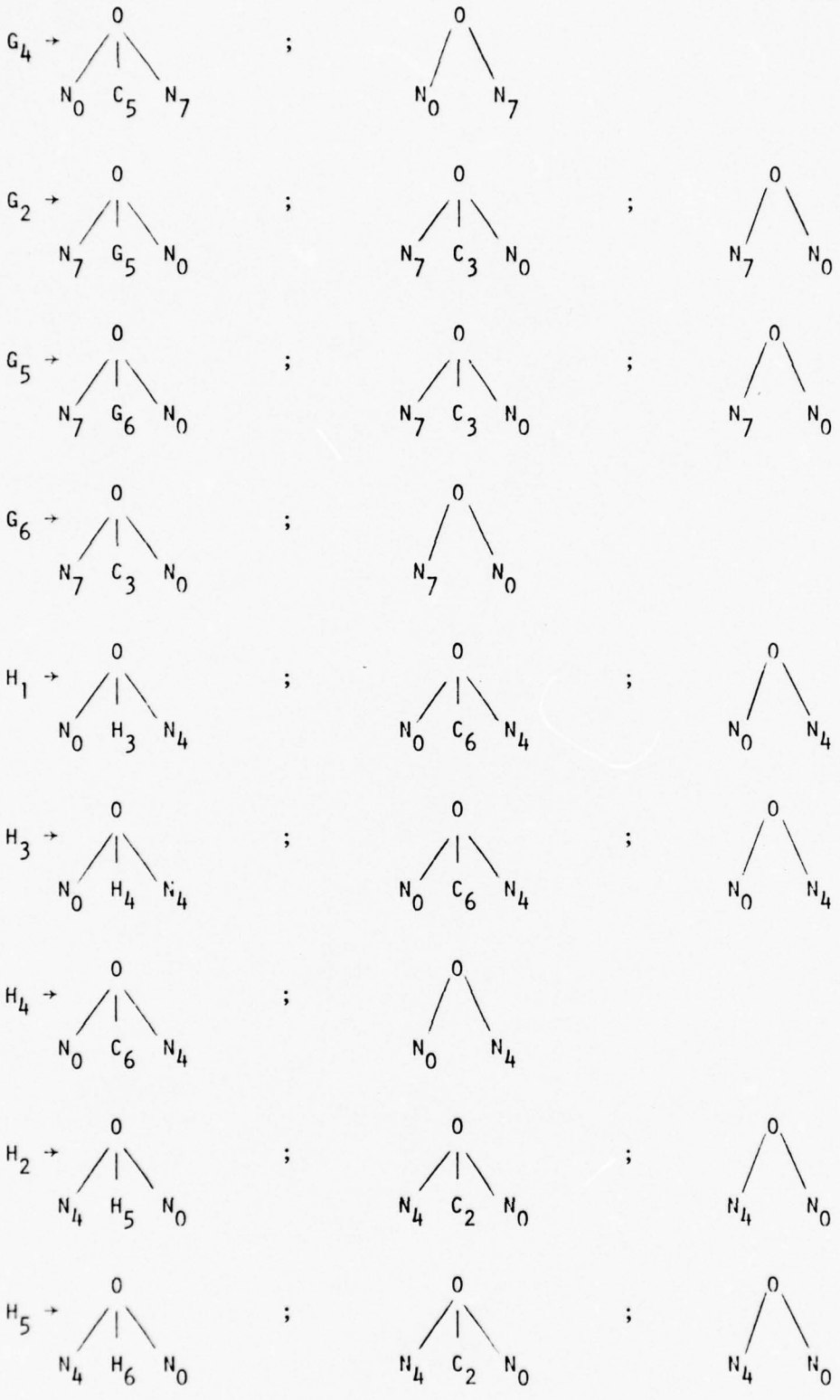
;

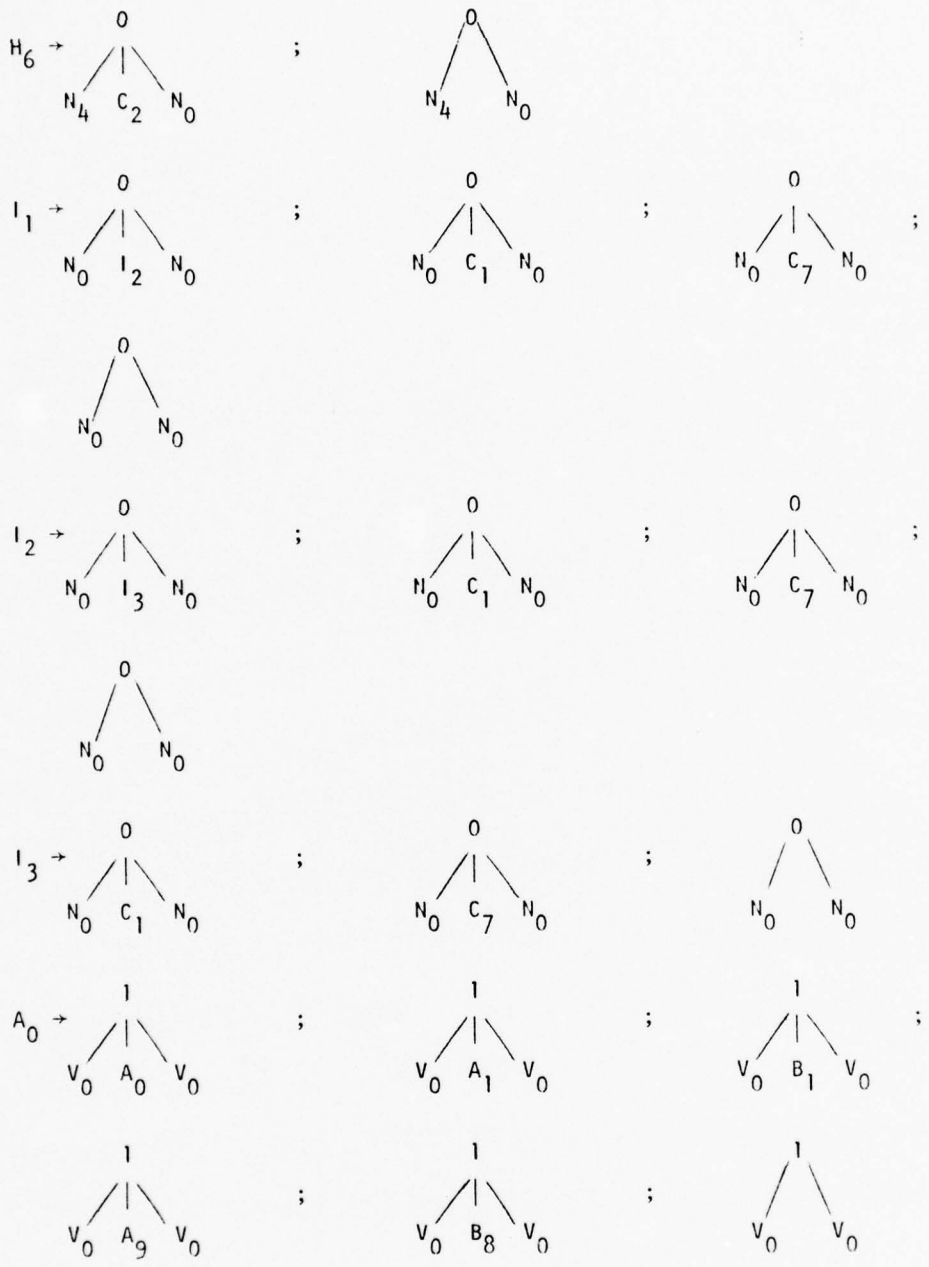


;


D4 → 1 N0 C2 V0 ; 1 N0 V0
D2 → 1 V0 D5 N0 ; 1 V0 C6 N0 ; 1 V0 N0
D5 → 1 V0 D6 N0 ; 1 V0 C6 N0 ; 1 V0 N0
D6 → 1 V0 C6 N0 ; 1 V0 N0
E1 → 0 N0 E3 V0 ; 0 N0 C3 V0 ; 0 N0 V0
E3 → 0 N0 E4 V0 ; 0 N0 C3 V0 ; 0 N0 V0
E4 → 0 N0 C3 V0 ; 0 N0 V0
E2 → 0 V0 E5 N0 ; 0 V0 C5 N0 ; 0 V0 N0
E5 → 0 V0 E6 N0 ; 0 V0 C5 N0 ; 0 V0 N0


E6 → O V0 C5 N0 ; O V0 N0
F1 → O N0 F3 V1 ; O N0 C4 V1 ; O N0 V1
F3 → O N0 F4 V1 ; O N0 C4 V1 ; O N0 V1
F4 → O N0 C4 V1 ; O N0 V1
F2 → O V1 F5 N0 ; O V1 C4 N0 ; O V1 N0
F5 → O V1 F6 N0 ; O V1 C4 N0 ; O V1 N0
F6 → O V1 C4 N0 ; O V1 N0
G1 → O N0 G3 N7 ; O N0 C5 N7 ; O N0 N7
G3 → O N0 G4 N7 ; O N0 C5 N7 ; O N0 N7


G4 → O N0 C5 N7 ; O N0 N7
G2 → O N7 G5 N0 ; O N7 C3 N0 ; O N7 N0
G5 → O N7 G6 N0 ; O N7 C3 N0 ; O N7 N0
G6 → O N7 C3 N0 ; O N7 N0
H1 → O N0 H3 N4 ; O N0 C6 N4 ; O N0 N4
H3 → O N0 H4 N4 ; O N0 C6 N4 ; O N0 N4
H4 → O N0 C6 N4 ; O N0 N4
H2 → O N4 H5 N0 ; O N4 C2 N0 ; O N4 N0
H5 → O N4 H6 N0 ; O N4 C2 N0 ; O N4 N0



Production rules with left-hand side nonterminals as $N_{0,1,\ldots,9}$ or $V_{0,1,\ldots,5}$ are the same as those in grammar G_5 in Appendix A-2.

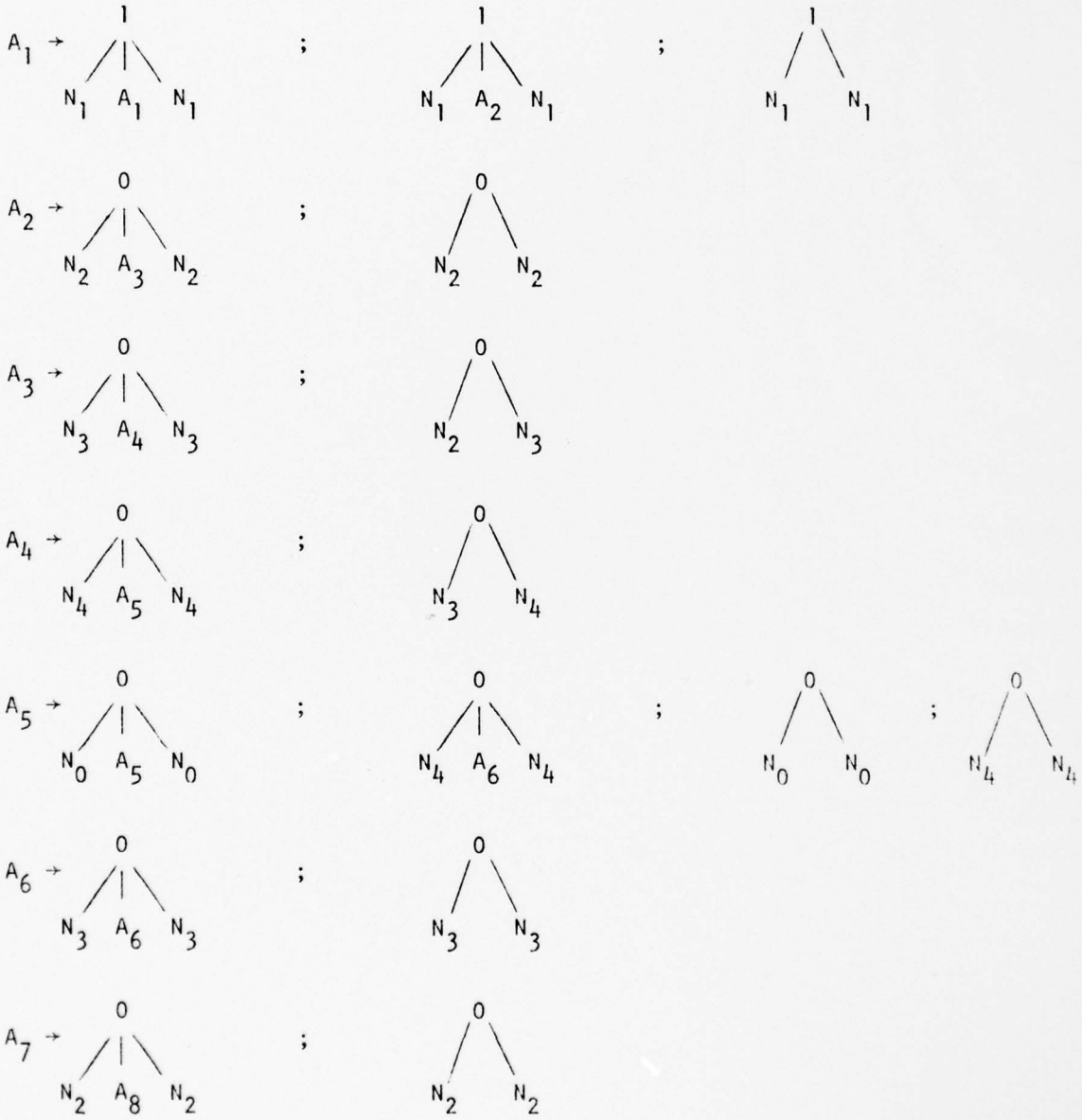
B-3. Discrimination Grammar for Pattern D34

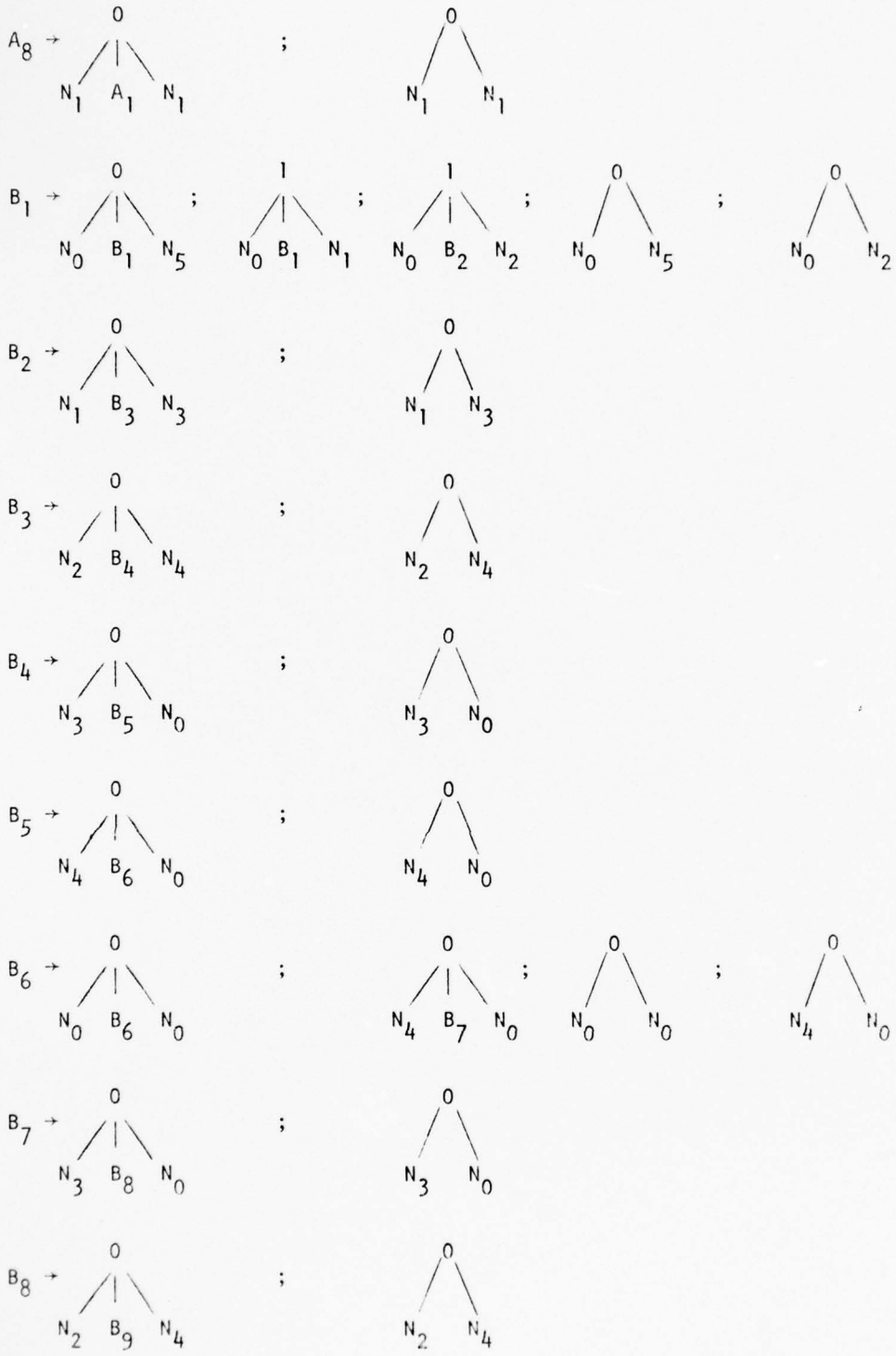
$G_{34} = (V_{34}, r, P_{34}, S_{34})$ over $<\Sigma, r>$

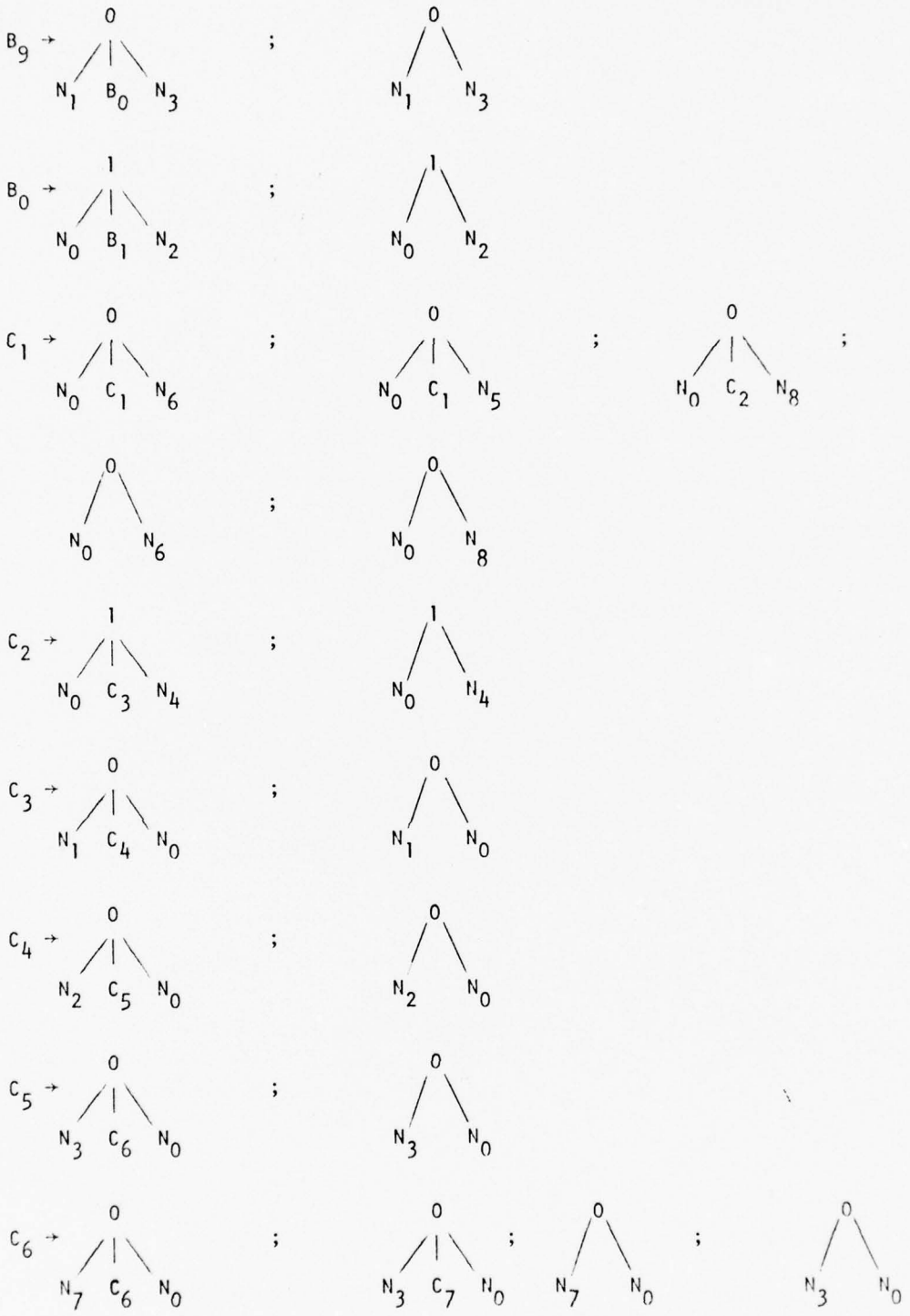
$V_{34} = S \cup \Sigma \cup N_{0,1,\ldots,6}$.

$S_{34} = \{A_{1,\ldots,8}, B_{0,1,\ldots,9}, C_{0,1,\ldots,9}, D_{0,1,\ldots,9}, E_{0,1,\ldots,9}, F_{0,1,\ldots,9}\}$

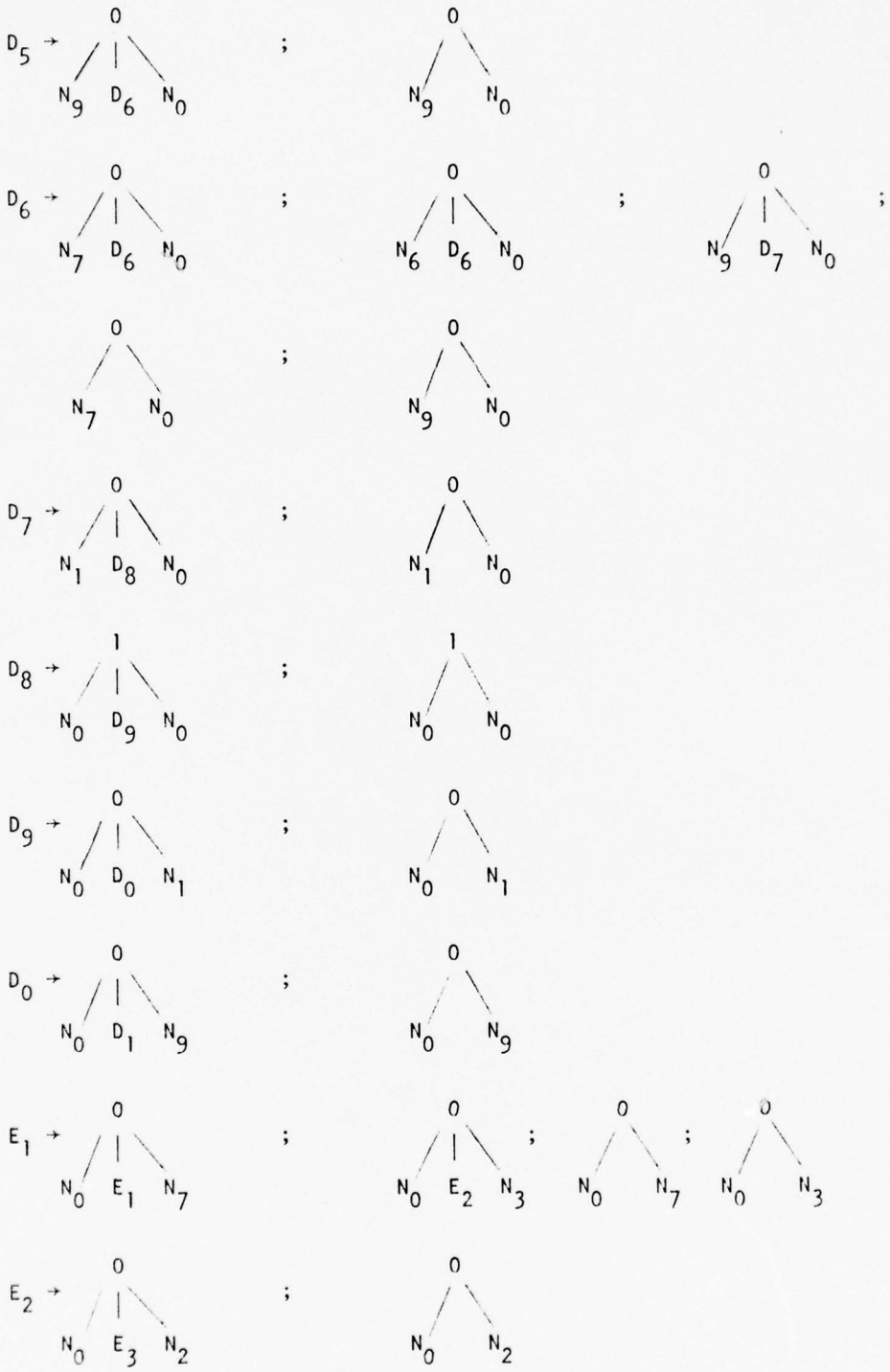
P_{34}:

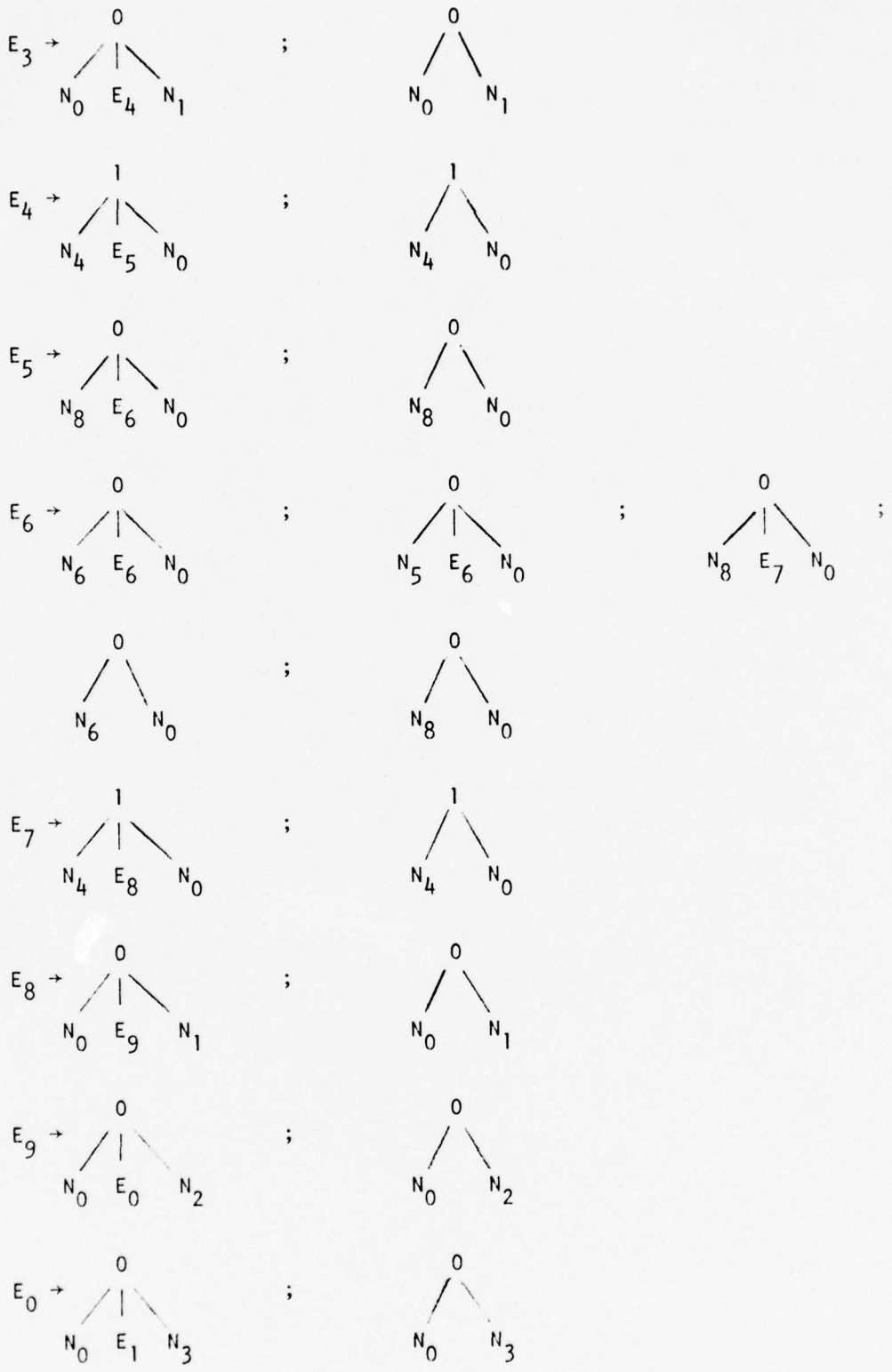


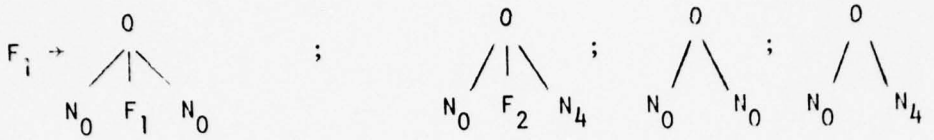

A8 → 0 N1 A1 N1 ; 0 N1 N1
B1 → 0 N0 B1 N5 ; 1 N0 B1 N1 ; 1 N0 B2 N2 ; 0 N0 N5 ; 0 N0 N2
B2 → 0 N1 B3 N3 ; 0 N1 N3
B3 → 0 N2 B4 N4 ; 0 N2 N4
B4 → 0 N3 B5 N0 ; 0 N3 N0
B5 → 0 N4 B6 N0 ; 0 N4 N0
B6 → 0 N0 B6 N0 ; 0 N4 B7 N0 ; 0 N0 N0 ; 0 N4 N0
B7 → 0 N3 B8 N0 ; 0 N3 N0
B8 → 0 N2 B9 N4 ; 0 N2 N4


B9 → 0 N1 B0 N3 ; 0 N1 N3
B0 → 1 N0 B1 N2 ; 1 N0 N2
C1 → 0 N0 C1 N6 ; 0 N0 C1 N5 ; 0 N0 C2 N8 ;
0 N0 N6 ; 0 N0 N8
C2 → 1 N0 C3 N4 ; 1 N0 N4
C3 → 0 N1 C4 N0 ; 0 N1 N0
C4 → 0 N2 C5 N0 ; 0 N2 N0
C5 → 0 N3 C6 N0 ; 0 N3 N0
C6 → 0 N7 C6 N0 ; 0 N3 C7 N0 ; 0 N7 N0 ; 0 N3 N0


C7 → 0 N2 C8 N0 ; 0 N2 N0
C8 → 0 N1 C9 N0 ; 0 N1 N0
C9 → 1 N0 C0 N4 ; 0 N0 N4
C0 → 0 N0 C1 N8 ; 0 N0 N8
D1 → 0 N0 D1 N7 ; 0 N0 D1 N6 ; 0 N0 D2 N9 ;
0 N0 N7 ; 0 N0 N9
D2 → 0 N0 D3 N1 ; 0 N0 N1
D3 → 1 N0 D4 N0 ; 1 N0 N0
D4 → 0 N1 D5 N0 ; 0 N1 N0


D5 → 0 N9 D6 N0 ; 0 N9 N0
D6 → 0 N7 D6 N0 ; 0 N6 D6 N0 ; 0 N9 D7 N0 ;
0 N7 N0 ; 0 N9 N0
D7 → 0 N1 D8 N0 ; 0 N1 N0
D8 → 1 N0 D9 N0 ; 1 N0 N0
D9 → 0 N0 D0 N1 ; 0 N0 N1
D0 → 0 N0 D1 N9 ; 0 N0 N9
E1 → 0 N0 E1 N7 ; 0 N0 E2 N3 ; 0 N0 N7 ; 0 N0 N3
E2 → 0 N0 E3 N2 ; 0 N0 N2


E3 → 0 N0 E4 N1 ; 0 N0 N1
E4 → 1 N4 E5 N0 ; 1 N4 N0
E5 → 0 N8 E6 N0 ; 0 N8 N0
E6 → 0 N6 E6 N0 ; 0 N5 E6 N0 ; 0 N8 E7 N0 ;
0 N6 N0 ; 0 N8 N0
E7 → 1 N4 E8 N0 ; 1 N4 N0
E8 → 0 N0 E9 N1 ; 0 N0 N1
E9 → 0 N0 E0 N2 ; 0 N0 N2
E0 → 0 N0 E1 N3 ; 0 N0 N3

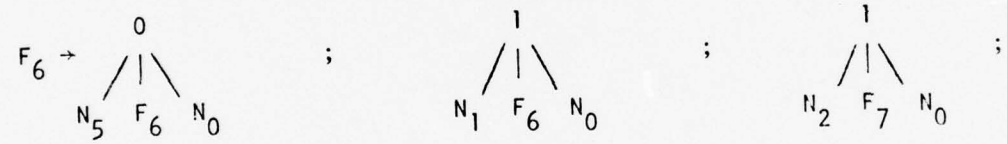

$F_i \rightarrow$ 0 N_0 F_1 N_0 ; 0 N_0 F_2 N_4 ; 0 N_0 N_0 ; 0 N_0 N_4


$F_2 \rightarrow$ 0 N_0 F_3 N_3 ; 0 N_0 N_3


$F_3 \rightarrow$ 0 N_4 F_4 N_2 ; 0 N_4 N_2


$F_4 \rightarrow$ 0 N_3 F_5 N_1 ; 0 N_3 N_1

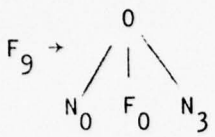
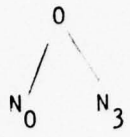

$F_5 \rightarrow$ 1 N_2 F_6 N_0 ; 1 N_2 N_0

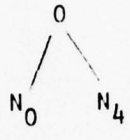

$F_6 \rightarrow$ 0 N_5 F_6 N_0 ; 1 N_1 F_6 N_0 ; 1 N_2 F_7 N_0 ;

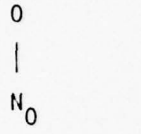

0 N_5 N_0 ; 1 N_2 N_0

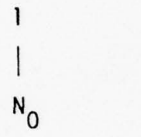
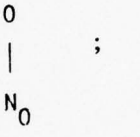

$F_7 \rightarrow$ 0 N_3 F_8 N_1 ; 0 N_3 N_1

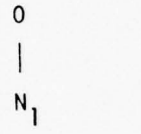

$F_8 \rightarrow$ 0 N_4 F_9 N_2 ; 0 N_4 N_2

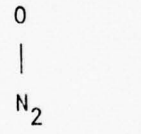
$F_9 \rightarrow$  ; 

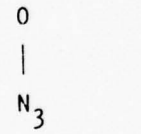
$F_0 \rightarrow$  ; 

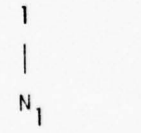
$N_0 \rightarrow$  ; 0

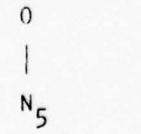
$N_1 \rightarrow$  ;  ; 1 ; 0

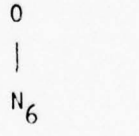
$N_2 \rightarrow$ 

$N_3 \rightarrow$ 

$N_4 \rightarrow$ 

$N_5 \rightarrow$ 

$N_6 \rightarrow$ 

$N_7 \rightarrow$ 

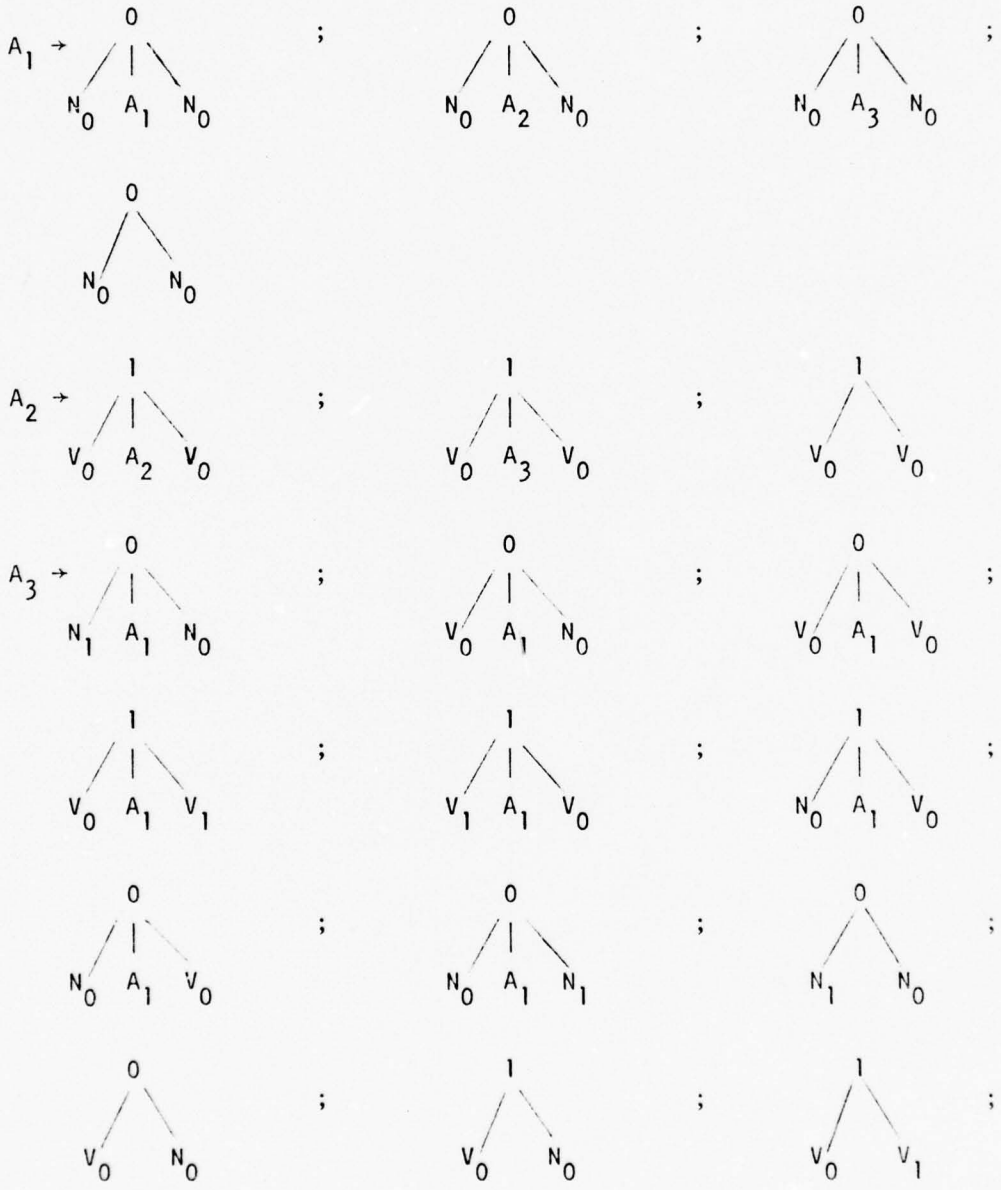
B-4. Discrimination Grammar for Pattern D68

$G_{68} = (V_{68}, r, P_{68}, S_{68})$ over $<\Sigma, r>$

$V_{68} = \{A_1, A_2, A_3, N_0, N_1, V_0, V_1\} \cup \Sigma$

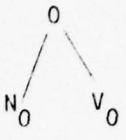
$S_{68} = \{A_1, A_2, A_3\}$

P_{68}:



;

;



;

$N_0 \rightarrow$ 0 | N_0 ; 0

$V_0 \rightarrow$ 1 | V_0 ; 1

$N_1 \rightarrow$ 0 | V_0 ; 0 | N_1

$V_1 \rightarrow$ 1 | N_0 ; 1 | V_1

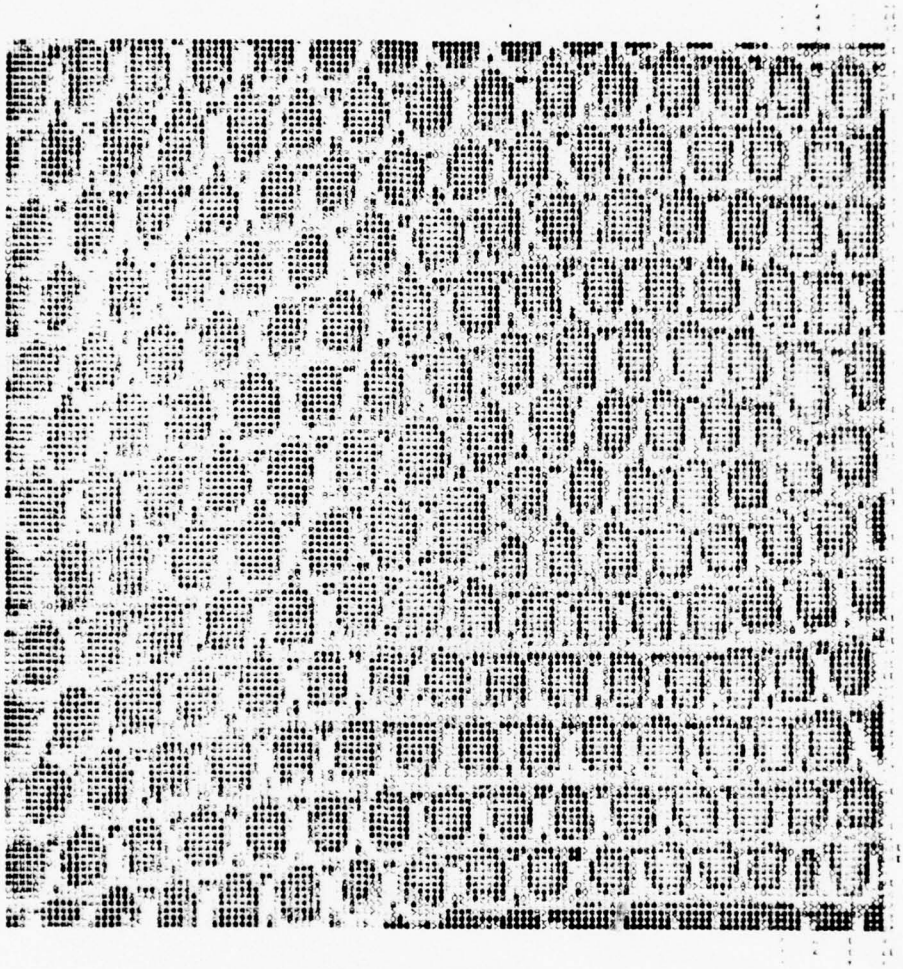


Fig. 1(a). Pattern D22. Reptile Skin.

Fig. 1. Four Texture Patterns obtained from Digitizing Pictures found in Brodatz's book, Textures.

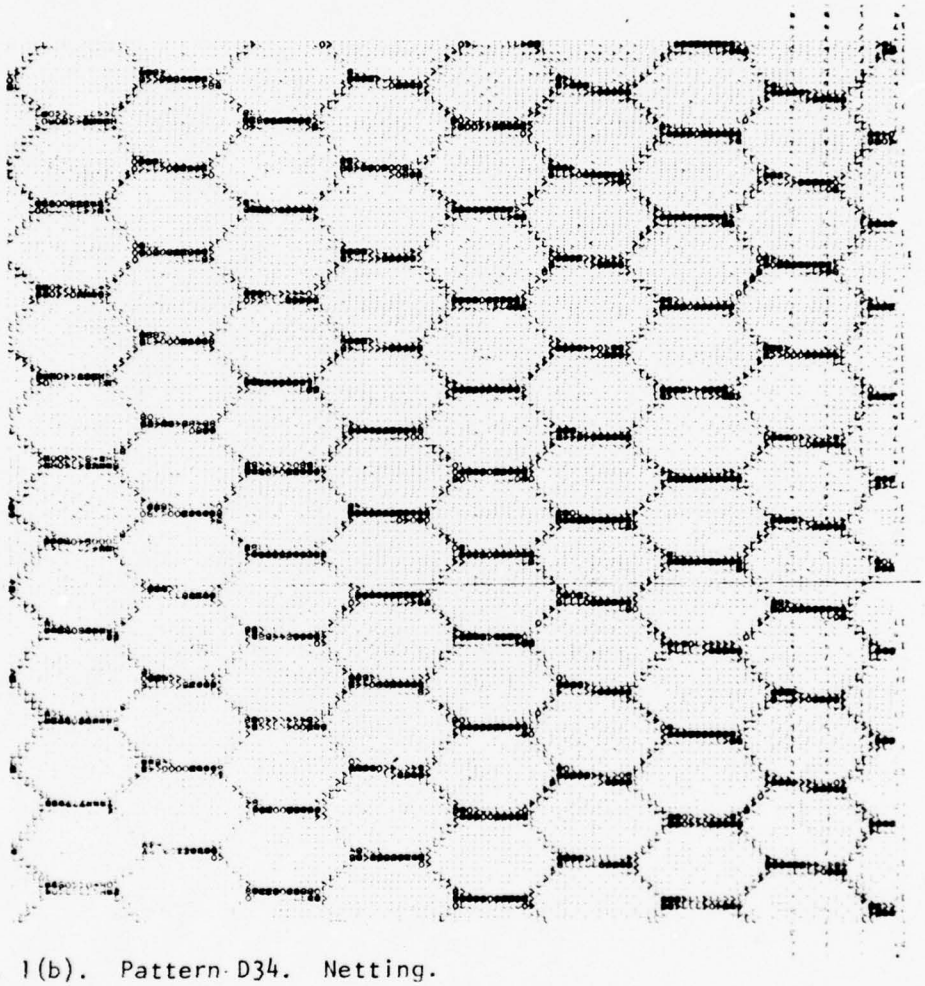


Fig. 1(b). Pattern D34. Netting.

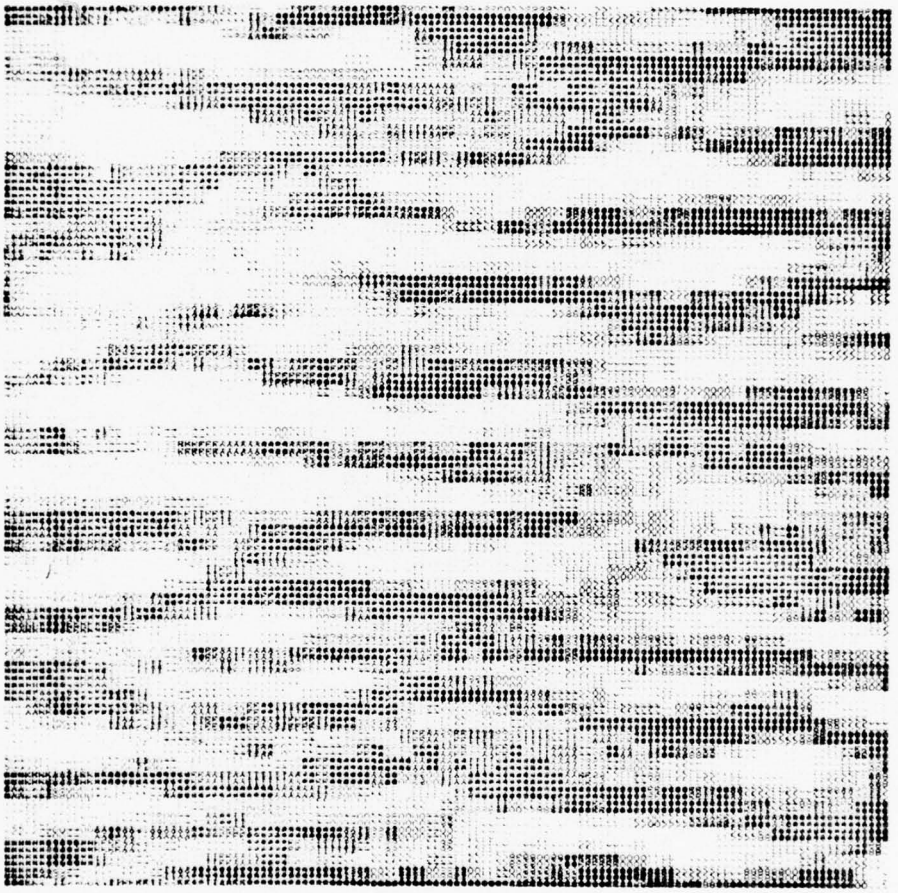


Fig. 1(c). Pattern D38. Water.

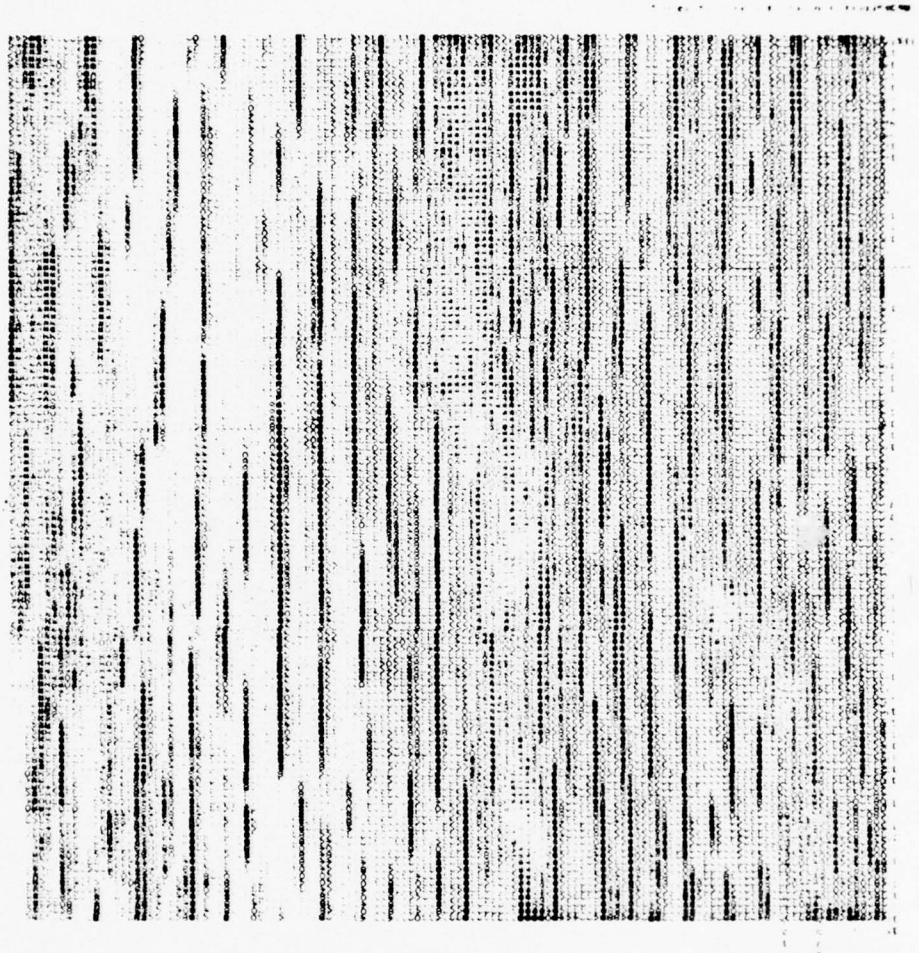
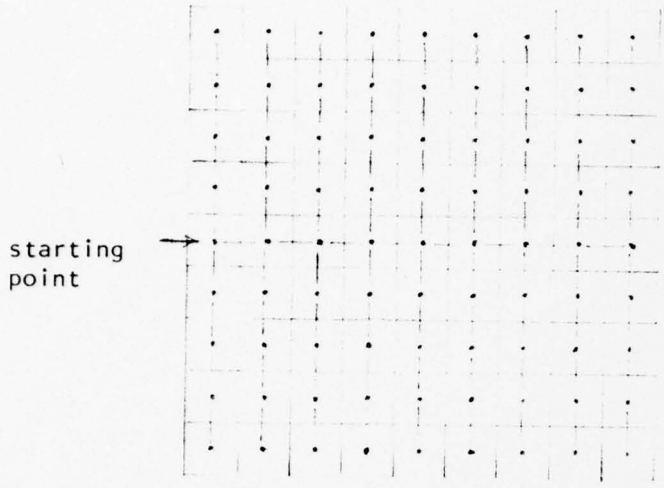
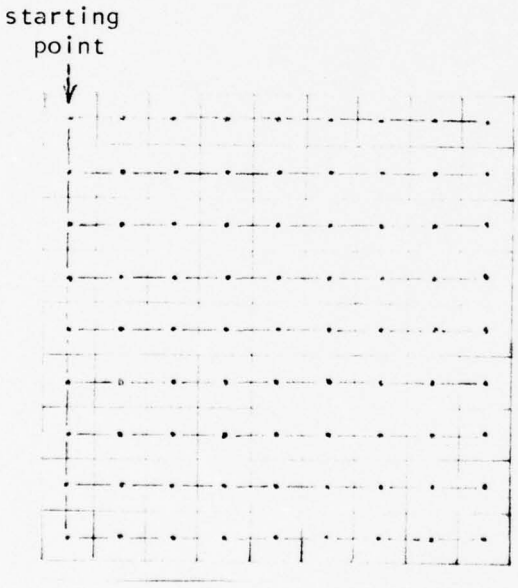


Fig. 1(d). Pattern D68. Woodgrain.

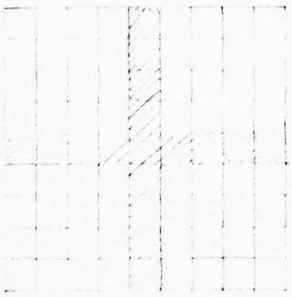


(a) Structure A

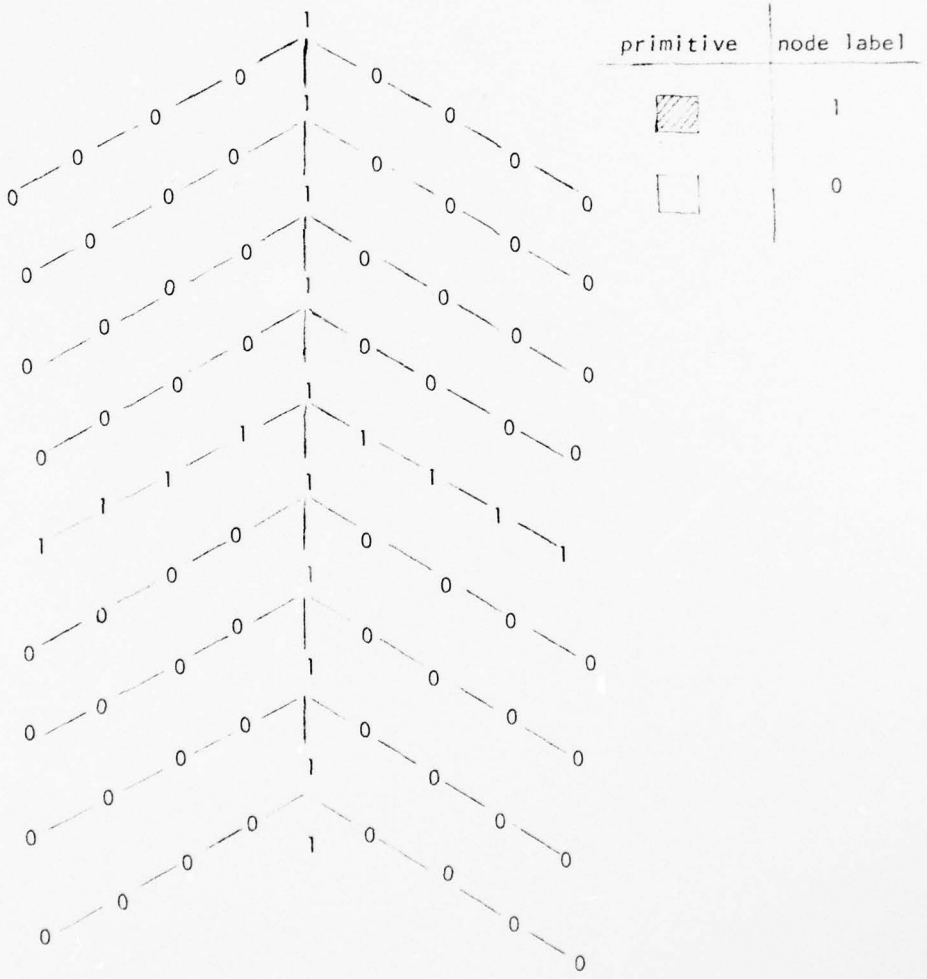


(b) Structure B

Fig. 2. Two tree structures for texture modeling.



(a)



(b)

Fig. 3. Pattern and its tree representation.

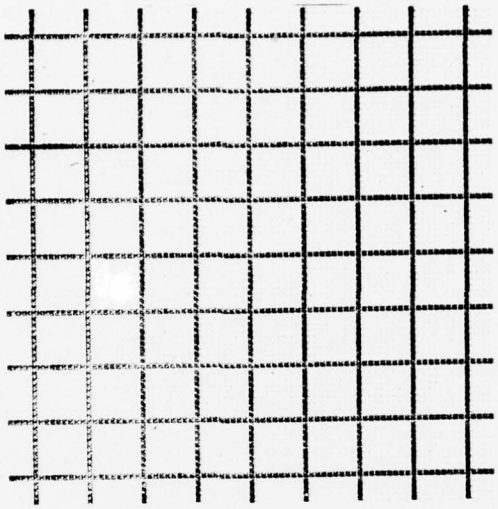


Fig. 4. A regular texture pattern.

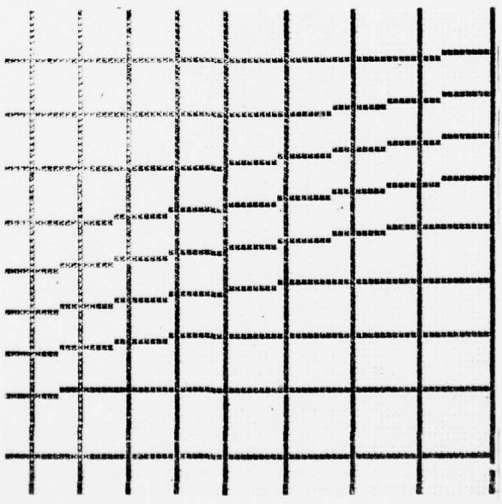


Fig. 5. A distorted texture pattern which can be generated by G_2.

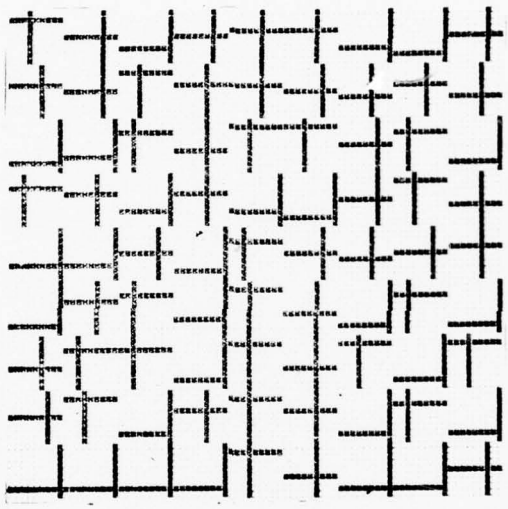


Fig. 6. A random texture pattern which could be generated by G_2.

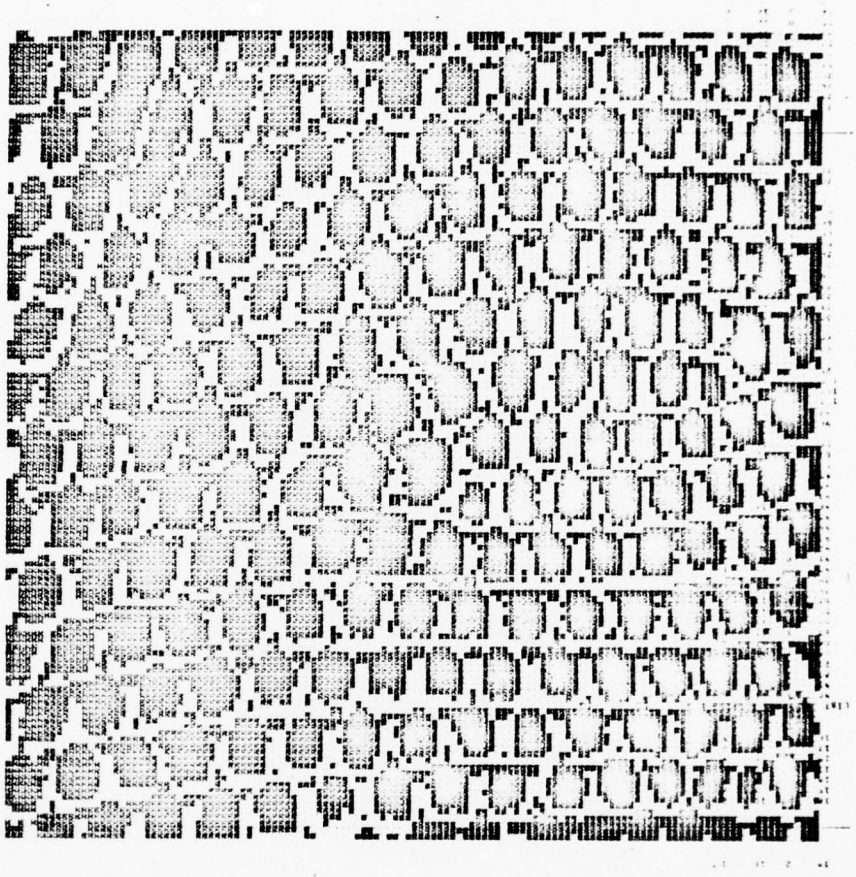


Fig. 7(a). Binary picture of pattern D22.

Fig. 7. Binary pictures of patterns in Fig. 1.

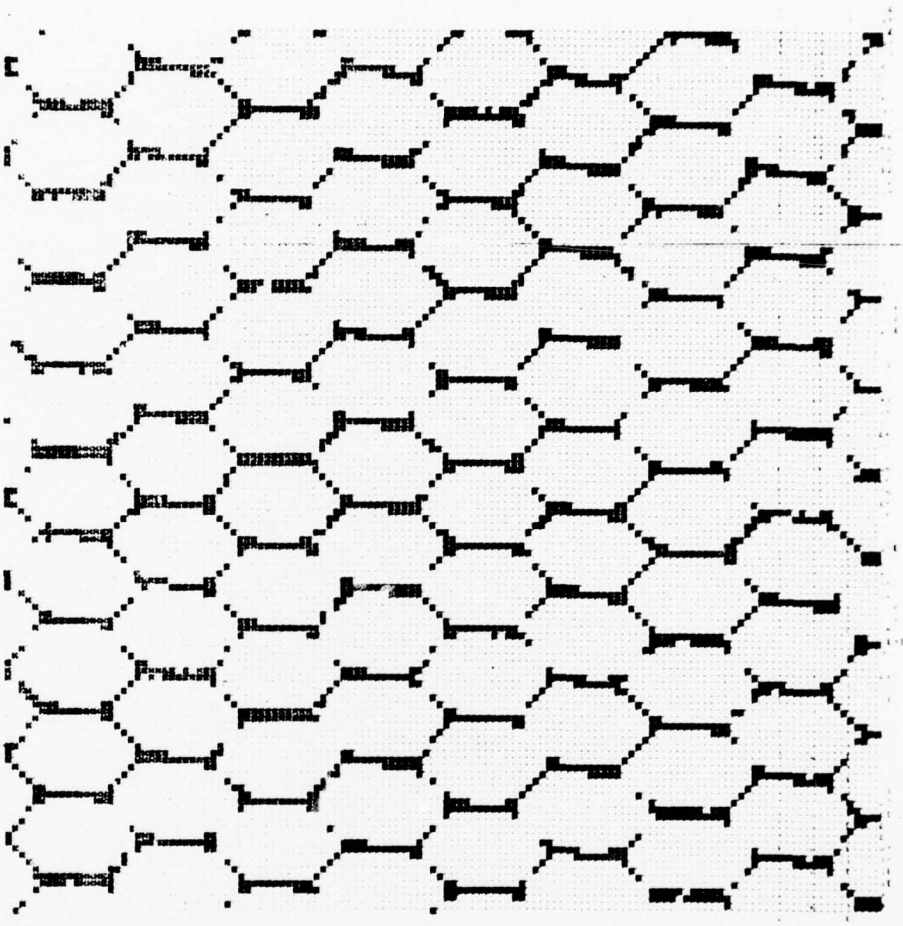


Fig. 7(b). Binary picture of pattern D34.

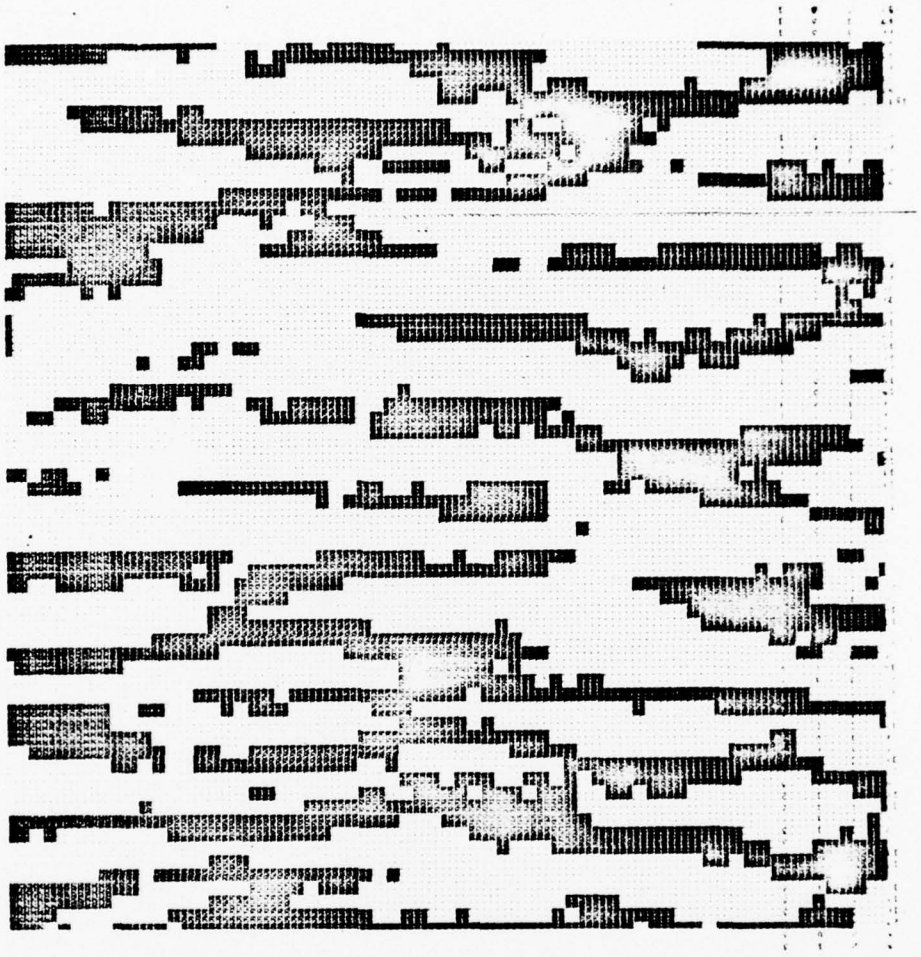


Fig. 7(c). Binary picture of pattern D38.

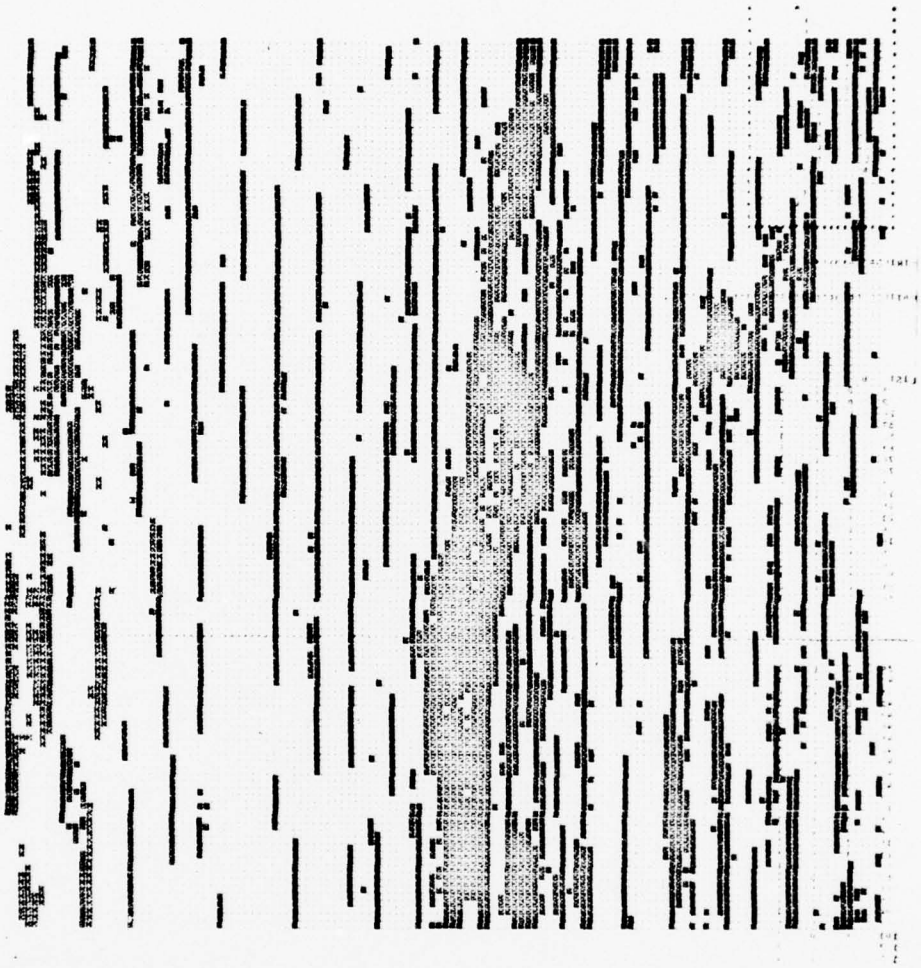
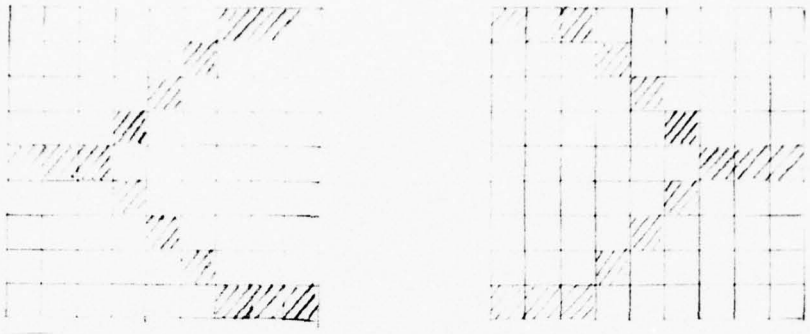


Fig. 7(d). Binary picture of pattern D68.



(a) (b)

Fig. 8. Windowed patterns of hexagonal tesselation.

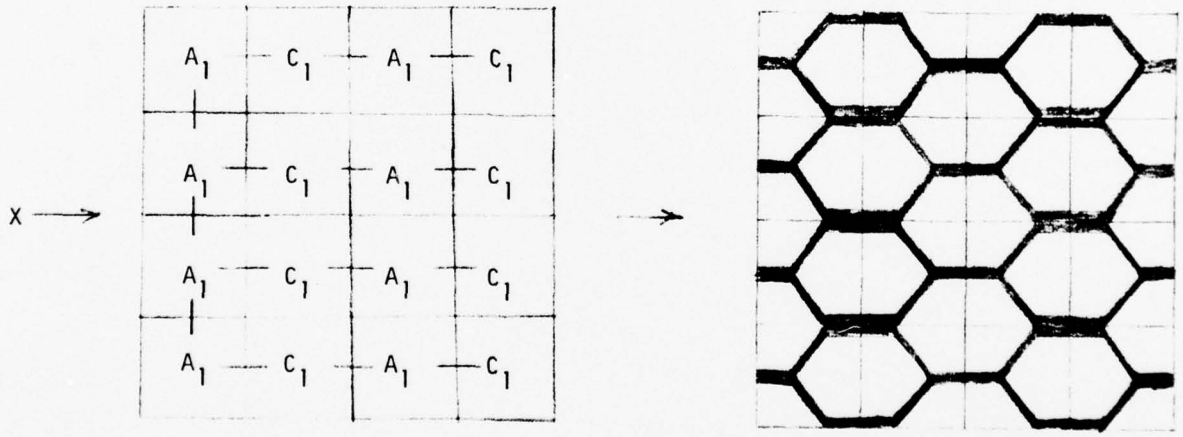


Fig. 9. The generation of a hexagonal pattern by using grammars G_3 and G_3'.

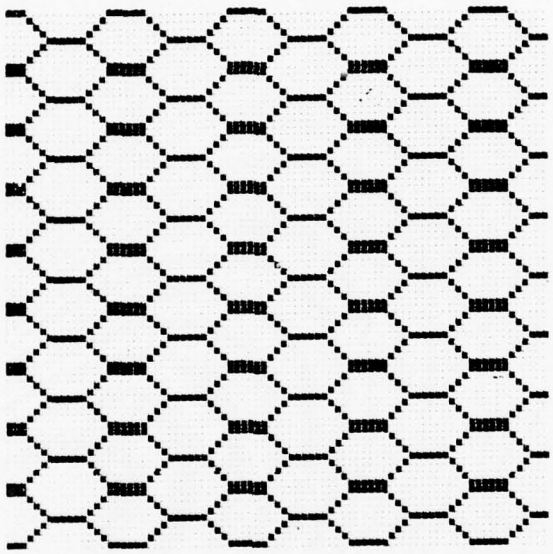
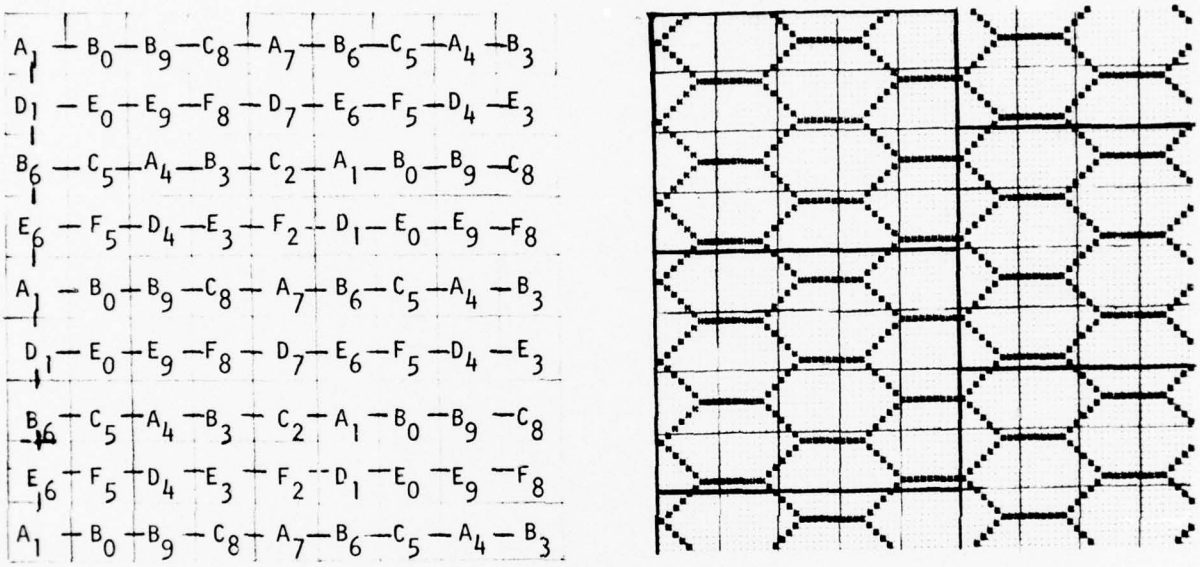


Fig. 10. The Generated Regular Hexagonals.



(a) Placement rule

(b) Result Pattern

Fig. 11. Generated Regular Hexagonals.

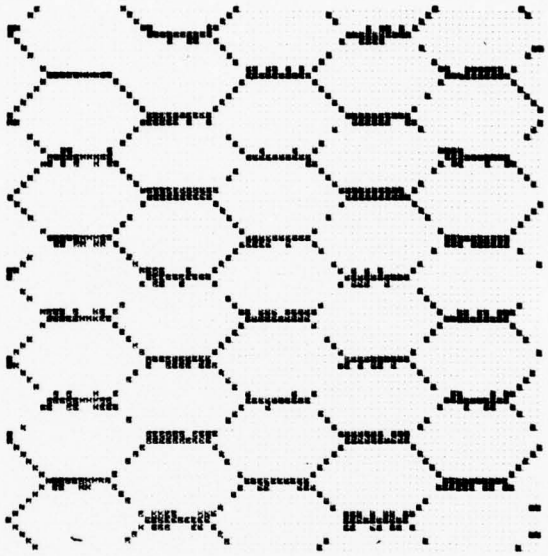


Fig. 12. Simulated Result of Pattern D34.

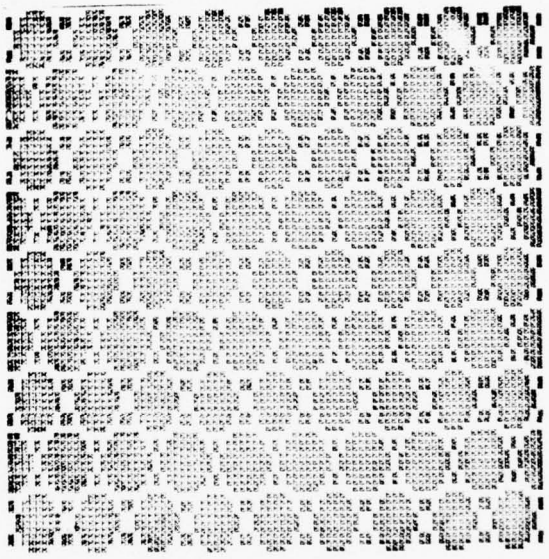
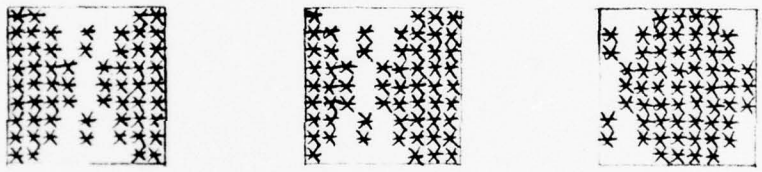


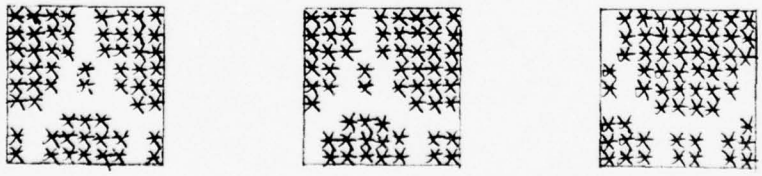
Fig. 13. The Ideal Texture of Pattern D34.

(a) (b)

Fig. 14. Basic pattern of Fig. 13.



A_1 A_2 A_5



D_1 D_2 D_5

Fig. 15. Windowed pattern primitives.

A_5	B_5	C_5	D_5	E_5	F_5	G_5	H_5	I_5
A_1	B_1	C_1	D_1	E_1	F_1	G_1	H_1	I_1
A_6	A_6	B_6	C_6	D_6	E_6	F_6	G_6	H_6
A_2	A_2	B_2	B_2	C_2	D_2	E_2	F_2	G_2
A_7	A_7	B_7	B_7	C_7	D_7	E_7	F_7	G_7
A_3	A_3	A_3	B_3	B_3	C_3	D_3	E_3	F_3
A_8	A_8	A_8	B_8	B_8	C_8	C_8	D_8	E_8
A_3	A_3	A_3	B_3	B_3	C_3	C_3	C_3	D_3
A_6	A_6	A_6	A_6	B_6	B_6	B_6	C_6	C_6

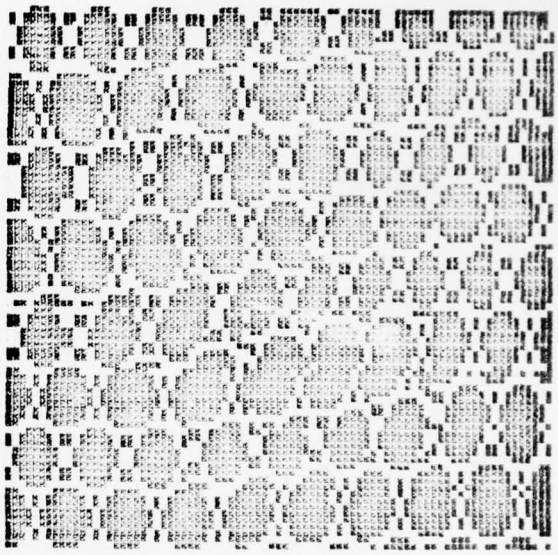


Fig. 16(a).

Fig. 16. Synthesis Results of pattern D22.

A_5	A_5	I_1	I_2	H_2	H_2	H_3	H_3	H_4
C_1	B_1	B_1	A_1	A_2	I_6	I_7	I_8	A_4
E_5	E_5	D_5	C_6	C_7	B_7	B_8	C_8	C_9
G_1	F_1	E_2	D_3	B_3	A_4	B_4	C_5	D_5
I_5	I_6	I_7	I_9	B_8	D_9	F_1	H_1	I_2
A_5	A_5	A_6	B_7	G_4	G_5	G_6	G_7	G_8
A_1	A_1	B_1	B_2	A_3	I_8	I_8	H_8	H_8
A_5	A_5	A_5	A_5	A_6	A_7	I_3	I_3	I_3
A_1	A_1	A_1	A_1	A_1	A_2	A_2	A_2	A_2

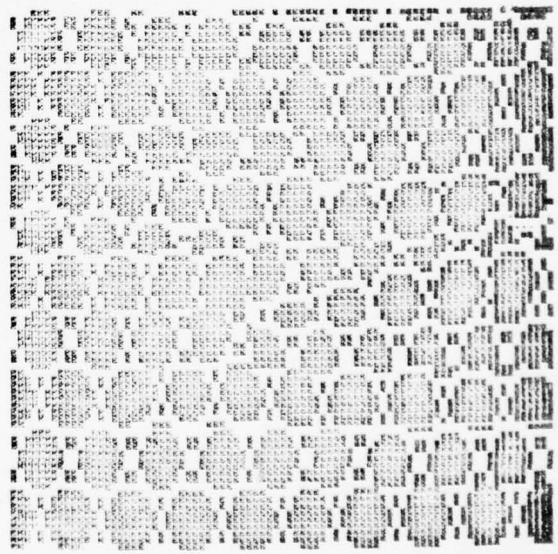


Fig. 16(b).

AD-A038 519 PURDUE UNIV LAFAYETTE IND SCHOOL OF ELECTRICAL ENGI--ETC F/G 12/2
A SYNTACTIC APPROACH TO TEXTURE ANALYSIS.(U)
FEB 77 S Y LU, K S FU AF-AFOSR-2661-74

UNCLASSIFIED TR-EE77-14 AFOSR-TR-77-0382 NL

2 OF 2
AD
A038 519

END
DATE
FILMED
5-77

H_4	H_4	I_5	A_1	A_1	I_7	H_8	H_8	H_9
G_6	H_7	I_8	A_5	A_5	I_3	H_4	G_5	G_5
F_3	G_3	H_4	A_1	A_1	H_8	G_9	F_9	F_9
F_6	G_7	H_7	A_5	A_5	H_4	G_4	F_5	F_5
E_3	F_3	H_3	A_1	A_1	H_9	F_9	E_9	E_9
D_6	F_6	H_7	A_5	A_5	H_4	F_5	D_5	C_5
C_1	E_3	G_3	A_1	A_1	G_9	E_9	C_9	B_9
C_1	E_7	G_7	A_5	A_5	G_4	E_4	C_4	B_4
C_3	E_3	G_3	A_1	A_1	G_9	F_9	C_9	A_9

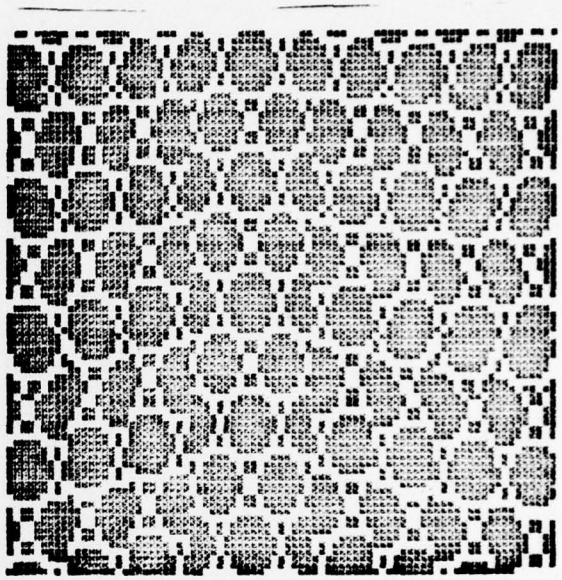
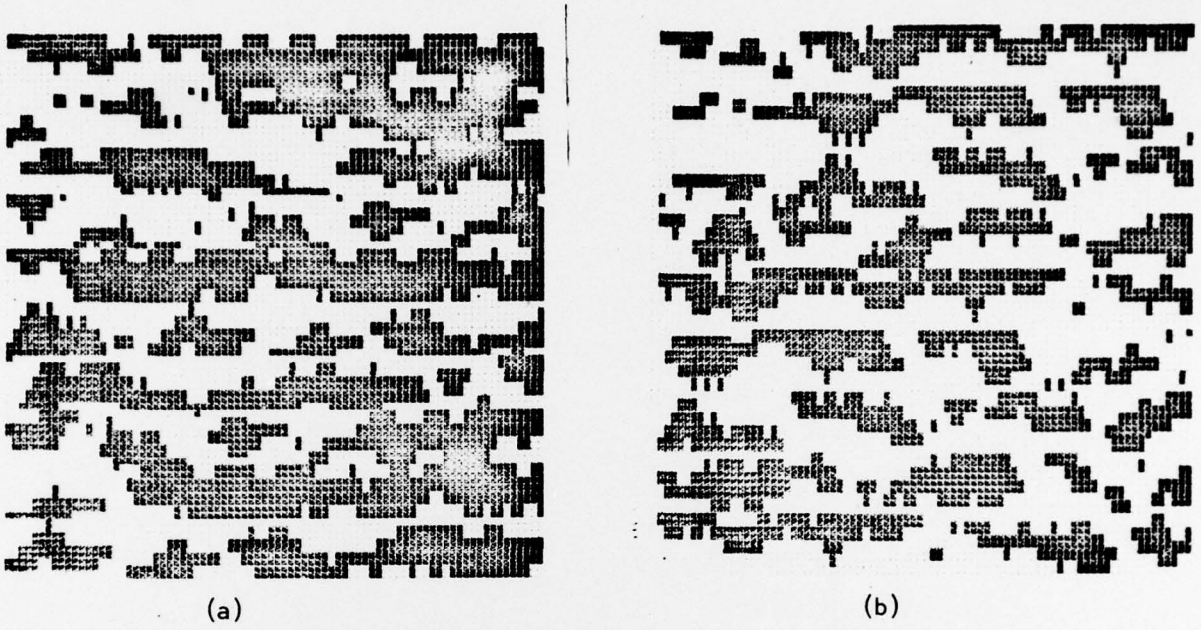
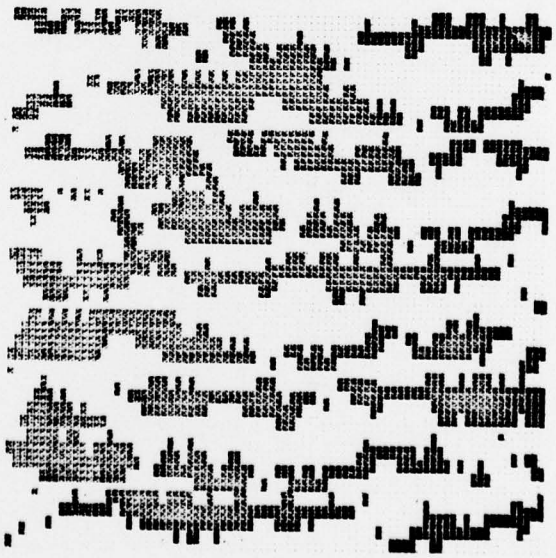


Fig. 16(c).

Fig. 17. The syntactic generation of water waves.

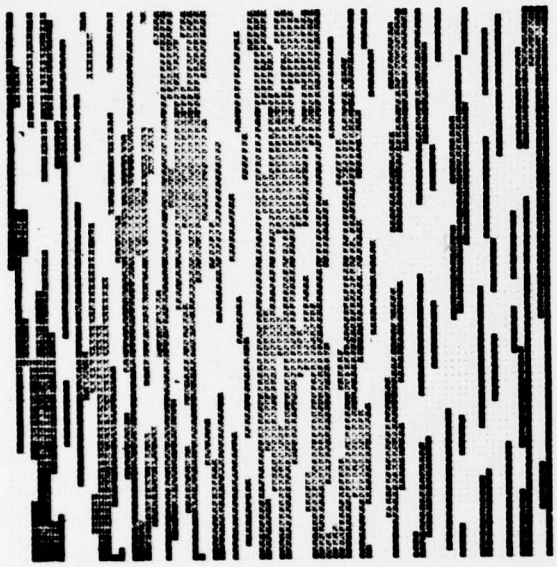


(a) (b)

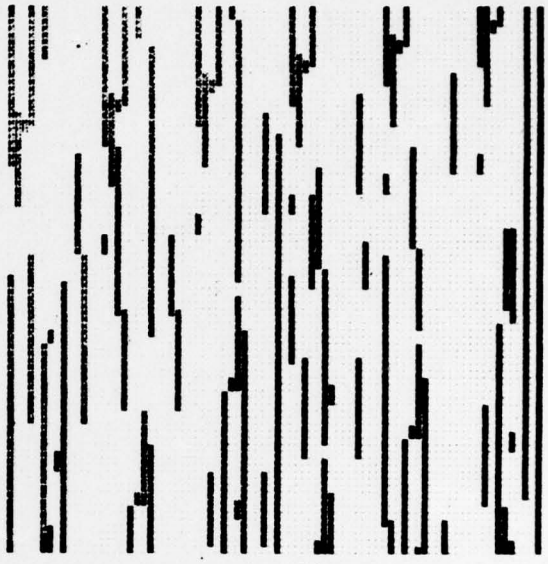


(c)

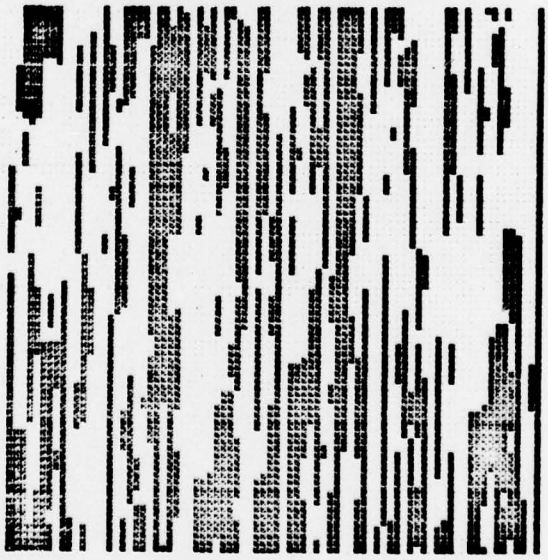
Fig. 18. Synthesis Results of Pattern D38.



(a)



(b)



(c)

Fig. 19. Synthesis results of Pattern D68.

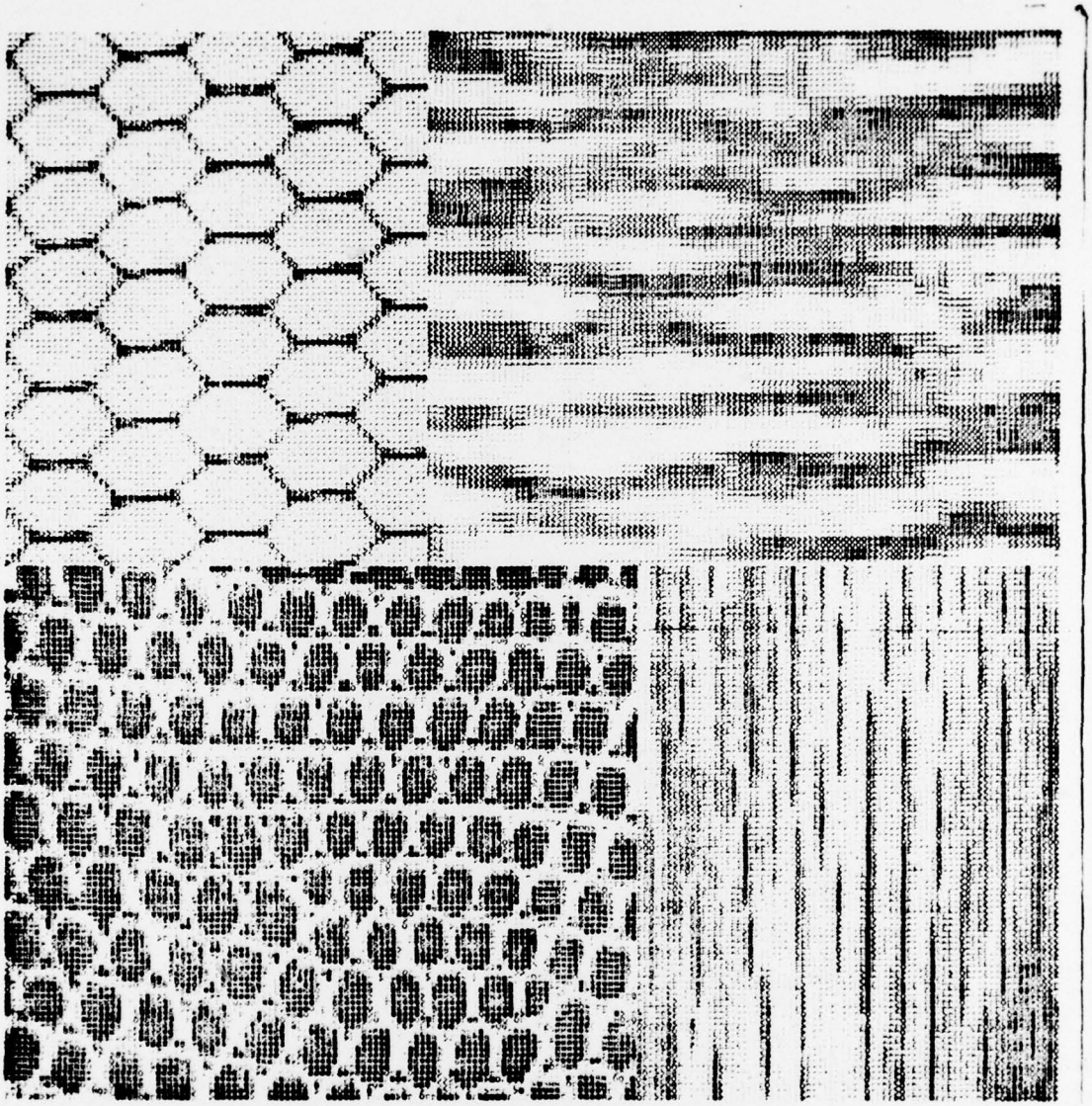


Fig. 20. Pictorial Data for Texture Discrimination.

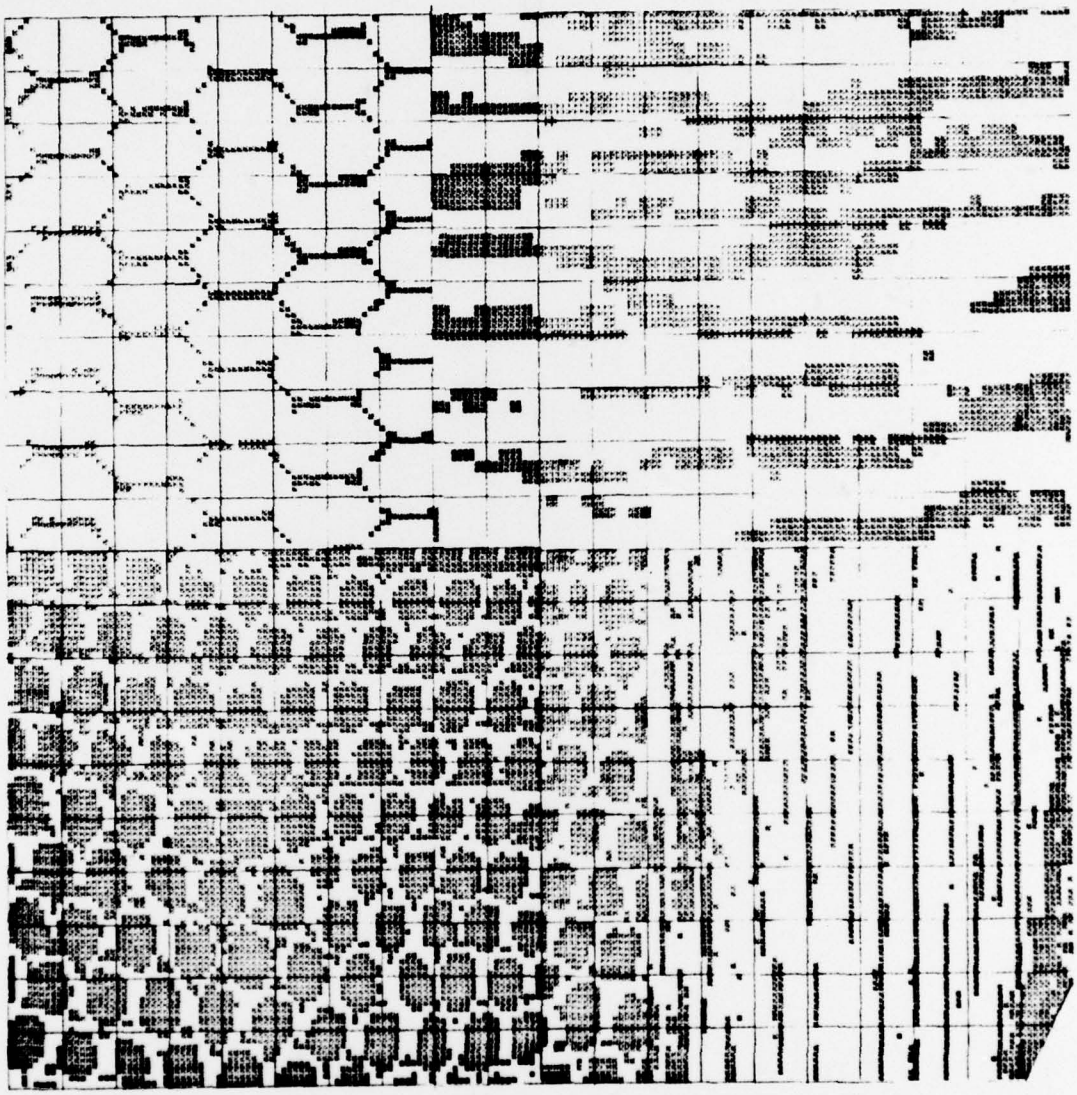
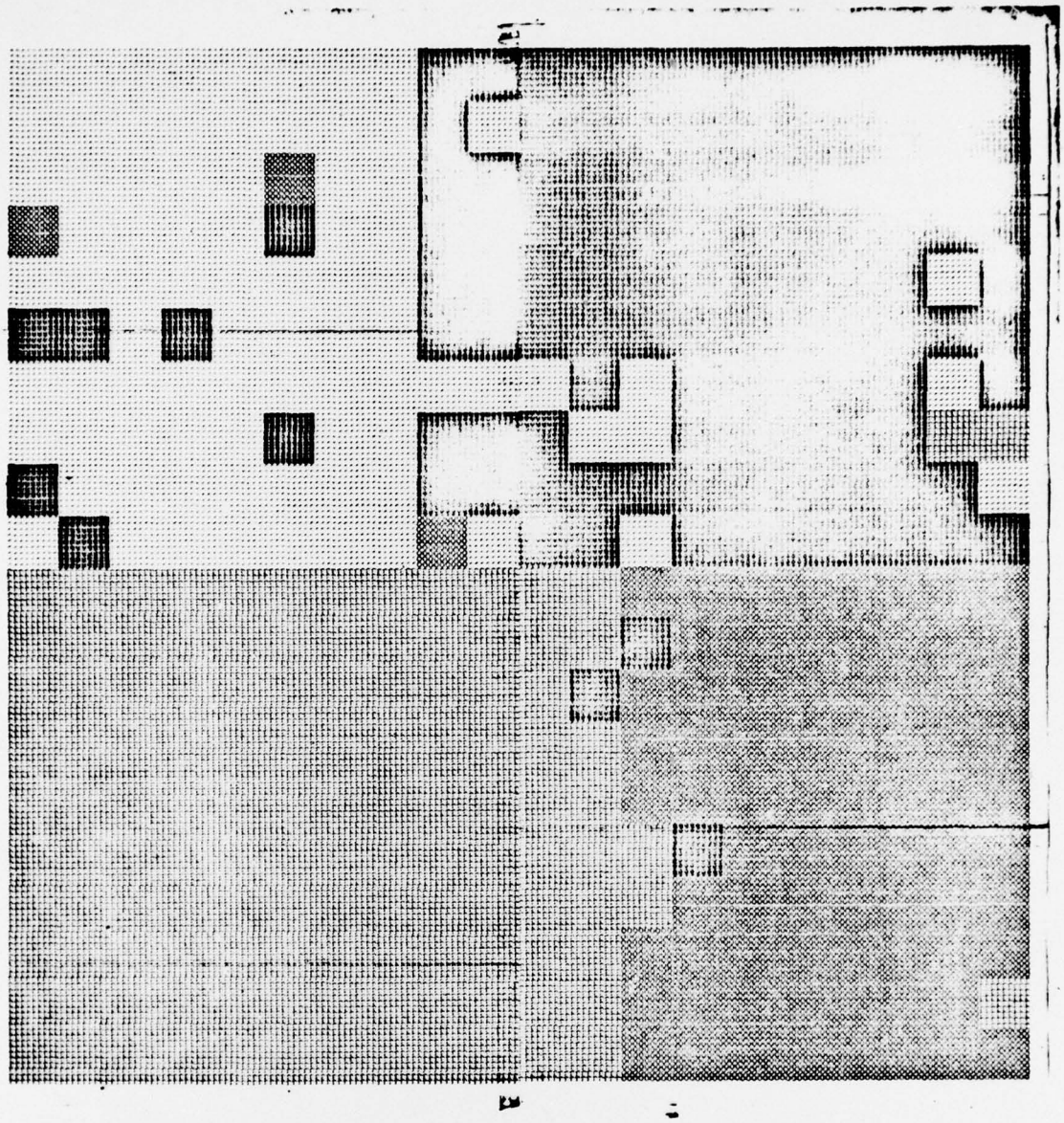
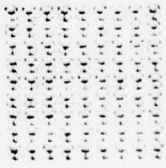


Fig. 21. Binary Picture of Fig. 20.



Color Code



Pattern D34 Pattern D22 Pattern D68 Pattern D38

Fig. 22. Discrimination Result.