Computers Viewing Artists at Work

by
Joan L. Kirsch and Russell A. Kirsch
The Sturvil Corp., Box 157, Clarksburg, MD, 20871, USA
and
Sanford Ressler
National Bureau of Standards
Gaithersburg, MD, 20899, USA

Reprinted from

Syntactic and Structural Pattern Recognition
eds: G. Ferrate, T. Pavlidis, A. Sanfeliu, H. Bunke
NATO ASI Series Vol. F45
Springer Verlag, Berlin, Heidelberg, 1988

Proceedings of the NATO Advanced Workshop on Syntactic and Structural
Pattern Recognition, held in Barcelona-Sitges, Spain, Oct. 23-25, 1986

COMPUTERS VIEWING ARTISTS AT WORK

J.L. Kirsch, R.A. Kirsch (*), S. Ressler(**)

(*)The Sturvil Corporation
P.O. Box 157
Clarksburg, MD 20871
(**) National Bureau of Standards
Gaithersburg, MD 20899
U.S.A.

Abstract

Our title suggests an Artificial Intelligence approach to the use of computers in the fine arts. We consider computers to have capabilities beyond the utilitarian ones of aiding in art making. Rather, we will investigate the possibility of computers seeing, even understanding, significant form in art. This understanding cannot rise autonomously, but must be the product of careful tutelage by artists, critics, and historians. A powerful tutorial mechanism to use for computers to learn about art is the picture grammar, which allows large classes of compositional structures to be described to a computer by the scholar who has a deep understanding of the art works. In this paper, we illustrate how a machine can be taught the compositional structure of the paintings of the contemporary artist Richard Diebenkorn. With such grammatical instruction, the computer can analyze existing paintings, generate new ones of the same style, and provide a beginning to a computational theory of style.

Formalism in art and computers

Computers are devices for manipulating symbols in formal systems. This characterization is broad enough to include uses ranging from numerical mathematics to computer graphics, although for graphics in art making, the formal properties must be supplemented by the physical characterization of output devices. But such a stretching of the definition is not necessary if we consider the use of computers in the formal description of art works. Here, the notion of formalism as it is understood in computer science and in art criticism come into reasonably close correspondence.

NATO ASI Series, Vol. F45
Syntactic and Structural Pattern Recognition
Edited by G. Ferraté et al.

The formal analysis of an art work deals with its hermetic visual properties, such as color, line, shape, materials and their arrangements, the so-called plastic elements. By contrast, there are extrinsic properties like feelings, stories, metaphor which are not so easily described to a computer. These are examples of what are often called the expressive properties of the art. Another common version of this distinction is the dichotomy between classical and romantic art. At this point, we can deal only with formal qualities and not extrinsic ones.

Actually, a bold body of criticism ranging from Roger Fry and Clive Bell to Clement Greenberg maintains that an aesthetic response to an art work depends solely on a purchase of its formal properties. Thus, describing the formal structure of art works to computers would seem a valid approach for making precise, to people and computers, the understanding of art.

Traditional methods of formal analysis have often been cumbersome. For example Loran[1] analyzed Cezanne's compositions with the use of schematic diagrams to account for space, planes, lines, volumes, and the other plastic elements. But a contemporary reader immediately realizes that such an analysis could be made more precise and complete with current computer graphic tools.

However still more powerful tools are available. They are drawn from the fields of image processing, pattern recognition, and computational linguistics, all parts of artificial intelligence. These tools arose [2] in 1964 when it was realized that the use of grammars for describing language could be generalized to describe images. Later, Stiny[3] further generalized these ideas, introducing the idea of a shape grammar. It is this form, slightly modified, that we have used in the study of painting.

Richard Diebenkorn's Ocean Park Paintings

Richard Diebenkorn is one of the most important and respected contemporary American painters. Between 1967 and 1986 he painted about 140 large oil paintings inspired by the look of the Ocean Park area of Santa Monica, California, where he has a studio. These painting appear, on first inspection, to be largely geometric and, hence, to lend themselves to the kind of formal analysis that we find most immediately possible. On further examination, one sees subtle and complex compositions built of the apparently informal and rich handling of layered colors, lines, and textures. Thus, a formal analysis restricted to the geometric qualities is, at best, a beginning of a complete grasp of the painting. It is this beginning that we have attempted.

A Diebenkorn Ocean Park Grammar

The grammar we have written [Kirsch 85] is a modified shape grammar. It contains 42 production rules, most of which offer options for how to subdivide regions of the painting. Some of these rules are recursive insofar as they produce subdivisions that, in turn, reinvoke the original rules. Such recursion enables the grammar to account for an infinite number of distinct compositions.

Although the grammar does not account for color, for example, it provides places where future rules for color may be added. This is in keeping with the practice, in computational linguistics, of providing "hooks" for the addition of features to an account of a corpus which is usually (in the case of language) arbitrarily large. In the case of Diebenkorn's painting, the complexity seems, at this early stage, arbitrarily large.

A shape grammar is a description of structure. For our grammar, it is a description of the linear compositional structure of this class of paintings. In any case, the grammar is equivocal with respect to two important possible uses. It may be used to analyze or to synthesize. The purpose of analysis is to reveal the structure of the paintings being described. Such an analysis can then be viewed by scholars to discover structure not necessarily evident in the finished work. The analysis is like the parse of a sentence. It conveys a significant part of the content of the object being analyzed. To see how such an analysis appears, Fig. 1 shows one of Diebenkorn's Ocean Park painting, number 111 of 1978. Fig . 2 shows the analysis assigned by the grammar to the painting. The rule numbers listed correspond to the grammar which assigns the analysis. Thus, for the grammar shown in the appendix we can see how some of the structural analysis is assigned to the painting by the grammar. We see several rules applied multiple times: rules 11,36,20, etc. These are some of the recursive rules. We also see how the grammar organizes the areas of the painting. Finally, we see how the highest level organization of the painting influences the lower levels. This is denoted by the notation /S that appears at the first level in OP/S, which denotes a kind of "suburban" landscape as opposed to the "rural" and "urban" alternatives provided by the grammar. In some of the final regions, like W/S we see the influence of this high level choice manifest in how the lower level regions remain, at this stage in the grammar. These properties are used, by later grammars, in choosing colors and other properties. These later properties represent more "surface" characteristics of the painting. The present grammar accounts for the "deep structure".

When the final analysis is produced by the grammar, it results in Fig. 3. We can compare this diagram with the original painting represented in Fig. 1, and see how the construction lines (the pentimenti) are both preserved by Diebenkorn, and generated by the grammar. This stage of the grammar only accounts for that part of the painting shown in Fig. 3. But it appears possible to "hang" further analyses on the superstructure assigned by this state of the grammar.

Testing the grammar

A grammar represents a theory, in this case, of the compositional structure of a class of paintings. There are a few ways to validate such a theory. The artist can be consulted to determine whether the grammatical analysis corresponds to his conscious plan of organization. There are perils in this approach, and of course, the uselessnes in the case of artists no longer living. Another form of validation consists of inspection by scholars with knowledge of the artist's oeuvre. They can compare their own intuitive analysis with that assigned by the grammar. Some day, it may even be possible for different grammars to be compared and evaluated, corrected and combined, just as computer programs are today. But the most powerful form of test available to us a present is the resynthesis test.

The grammar is a description of structure. Separate algorithms must be written to perform analysis with respect to the grammar. But algorithms can also be written which will synthesize pictures ab initio from the grammar. These pictures need not correspond at all to any extant paintings. In fact, since the grammar accounts for an infinite number of paintings, it is highly unlikely that a picture generated at random from the grammar will correspond to one that actually exists. But all such pictures have the structure that the grammar allows. We can exploit this property to test the grammar. By generating a random picture (making random choices when the grammar provides alternatives) we can inspect the final pictures for similarity with the artists oeuvre.

The picture of Fig. 4 is such a pseudo-Diebenkorn. It was (randomly) chosen to be a suburban landscape just as is that of Fig. 3. It thus has both a "busy" and "open" region. When Richard Diebenkorn saw this randomly generated picture he had "the immediate shock of recognition", notwithstanding the unusual history of its construction. The reader can with the grammar [Kirsch 85] in hand, perform such tests himself, testing the product with his own intuitive ability to recognize Diebenkorn's Ocean Park paintings.

Implementation

An additional testing mechanism is to implement the grammer. The rigours of producing a running system provide some interesting lessons. This particular implementation is concerned only with using the grammar to generate pseudo-Diebenkorns. One can control the system in a variety of ways. The rules are selected at random from among the valid ones at a given point in the production. Because many of the rules are recursive we had to weight the randomness towards the non-recursive rules so the program didn't always get trapped in recursion. Figure 5 explains the layout of the system and some of the capabilities.

The system can produce primarily two types of images. One is simply a single finished picture (fig 6) and the other is a series of images where each image represents the application of a single rule culminating in the finished picture (fig 7).

Fig.1 : Richard Diebenkorn, Ocean Park No. 111, 1978, Oil and Charcoal on Canvas, 336.2 x 336.7 cm , Courtesy Hirshhorn Museum and Sculpture Garden, Smithsonian Institution.

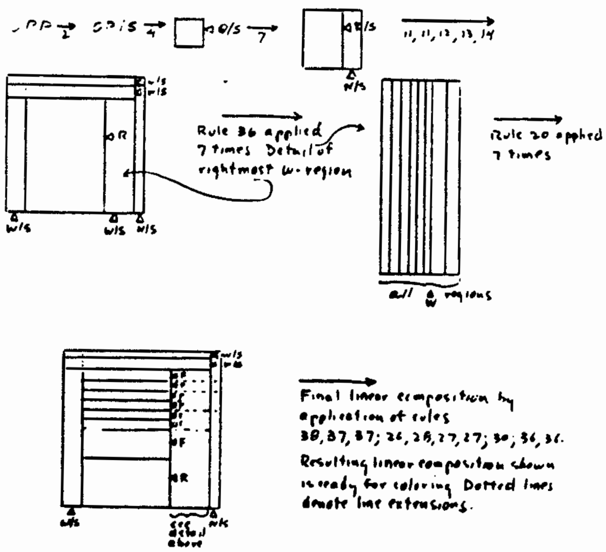
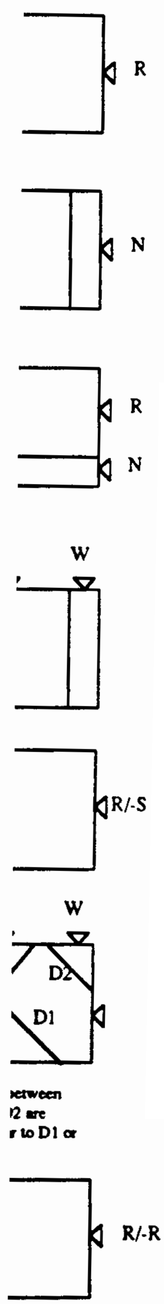


Fig.2 : Grammatical derivation of linear composition for Diebenkorn's Ocean Park No. 111. Rule numbers below arrows.

endix



between
2 are
r to D1 or

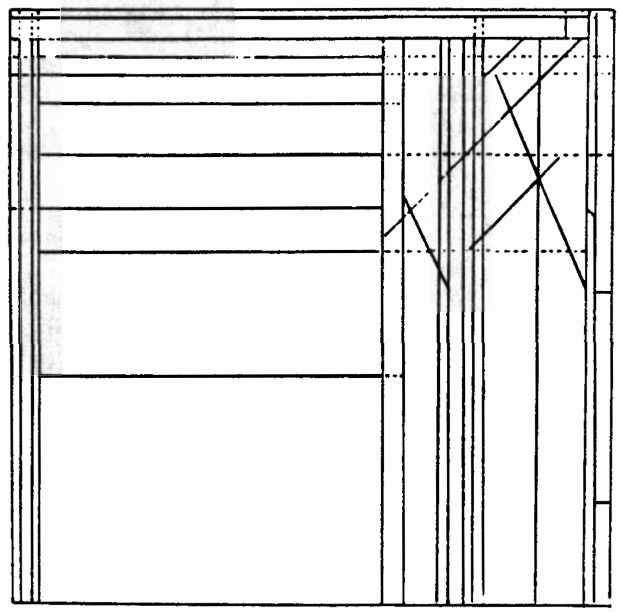


Fig.3 : Linear composition of Ocean Park No. 111.

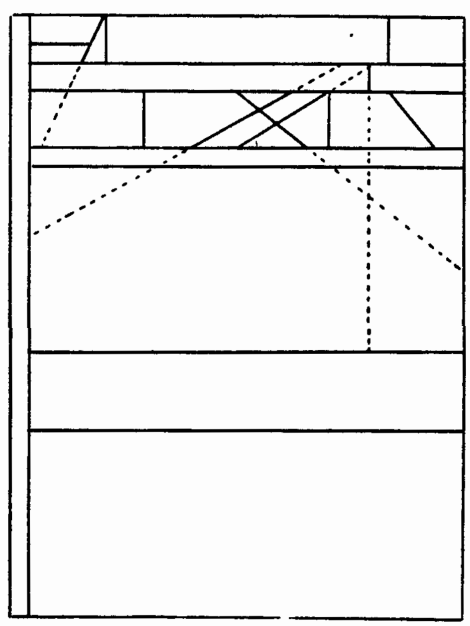


Fig.4 : A pseudo-Dibenkorn derived from the following sequence of rule applications : 2, 6, 17, 17, 11, 31, 31, 31, 30, 38, 37, 30, 31, 30, 30, 30, 32.

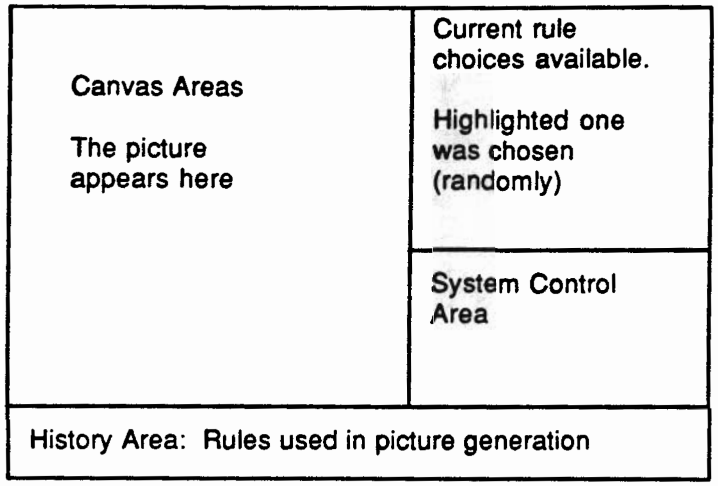


figure 5

After a picture has been generated the bottom row of the window displays a series of icons which represent the history of the rules used in the generation. More interestingly you may select the history rule and then see the geometry in the picture which corresponds to that rule. (It flashes) A extentions to the system would be to be able to modify the rule dynamically and then see the change in the picture as it propogates down the geometry. Such interactive methods may prove to be valuable in generating the grammar in the first place.

References

1. Erle Loran, "Cezanne's Compositions", Berkeley, Ca, University of California Press, 1943.
2. Russell A. Kirsch, "Computer Interpretation of English Text and Picture Patterns", IEEE Trans. Elect. Computers, EC13 (Aug 1964), p. 363.
3. George Stiny, "Introduction to Shape and Shape Grammars", Environment and Planning B, 7:(1980) p. 343.
4. Joan L. Kirsch, Russell A. Kirsch, "The Structure of Paintings: Formal Grammar and Design", Planning and Design 13:2(1986).

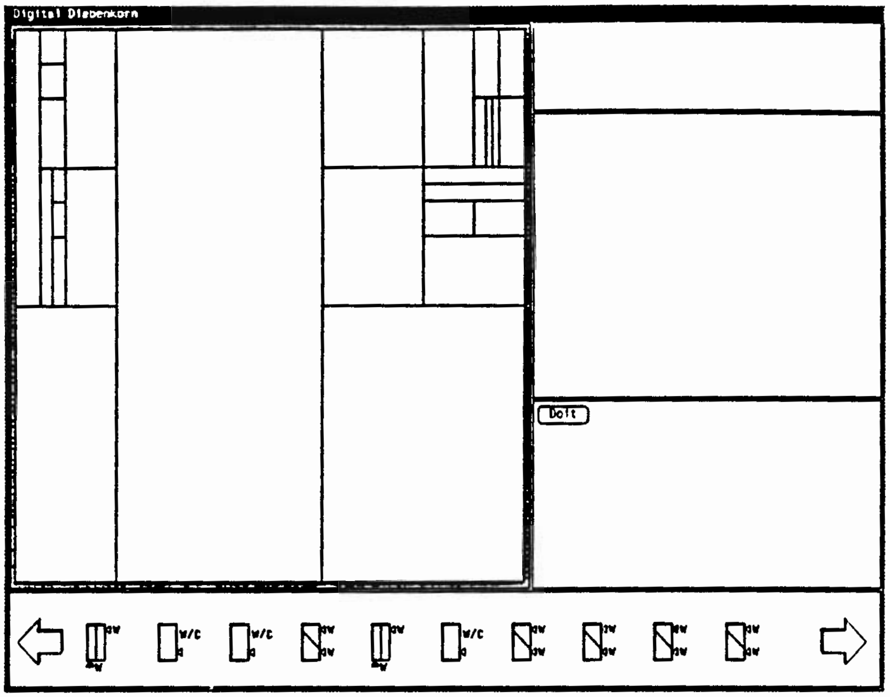


figure 6

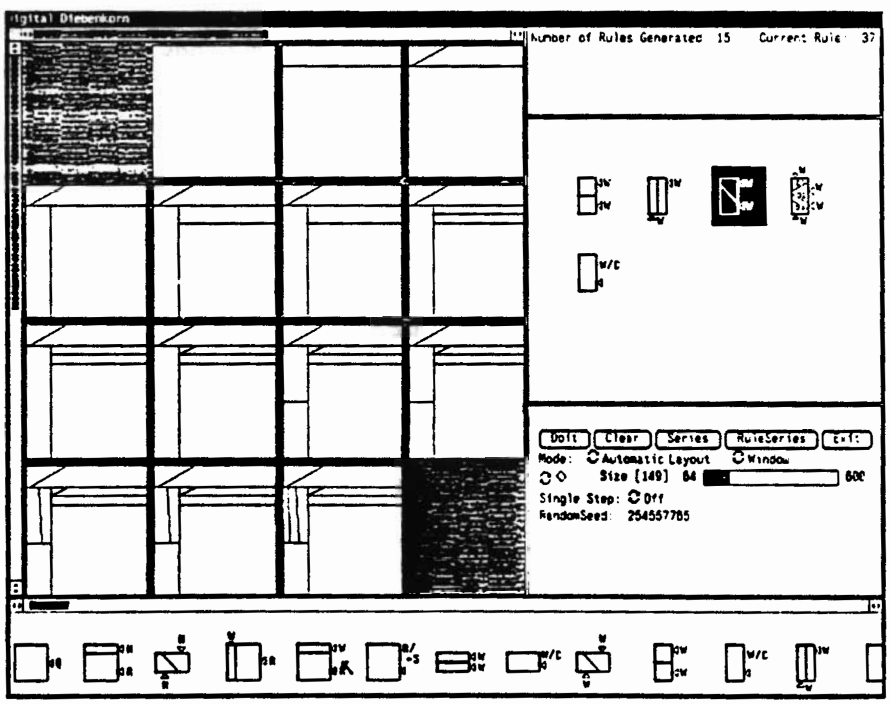


figure 7

Ocean Park grammar rules

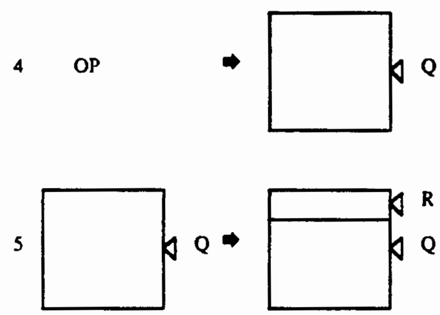
Appendix

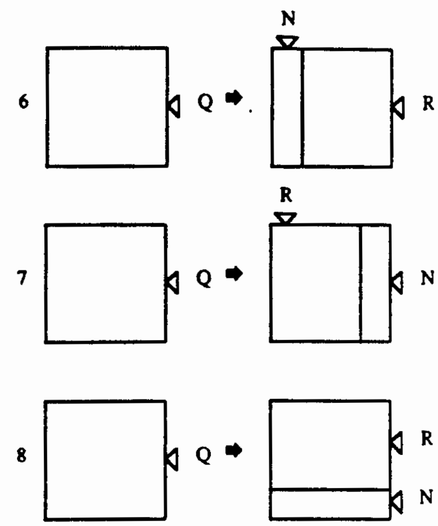
1 OPP ➡ OP/U

2 OPP ➡ OP/S

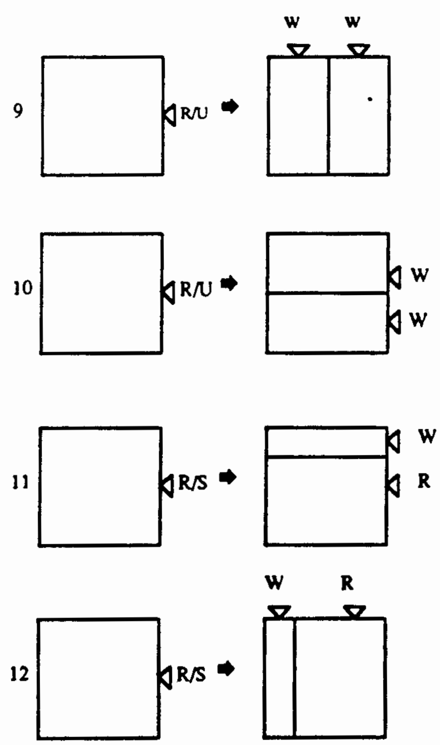
3 OPP ➡ OP/R

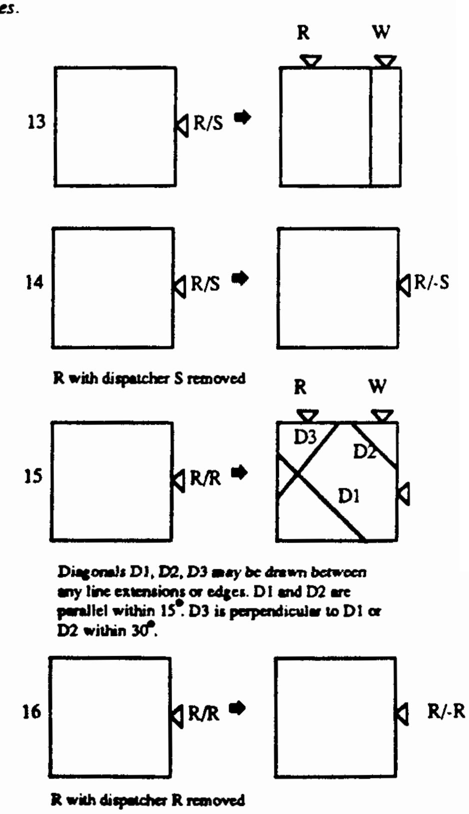
Dispatcher properties (U,S,R, etc) are retained (by default) for all constituents of a rule



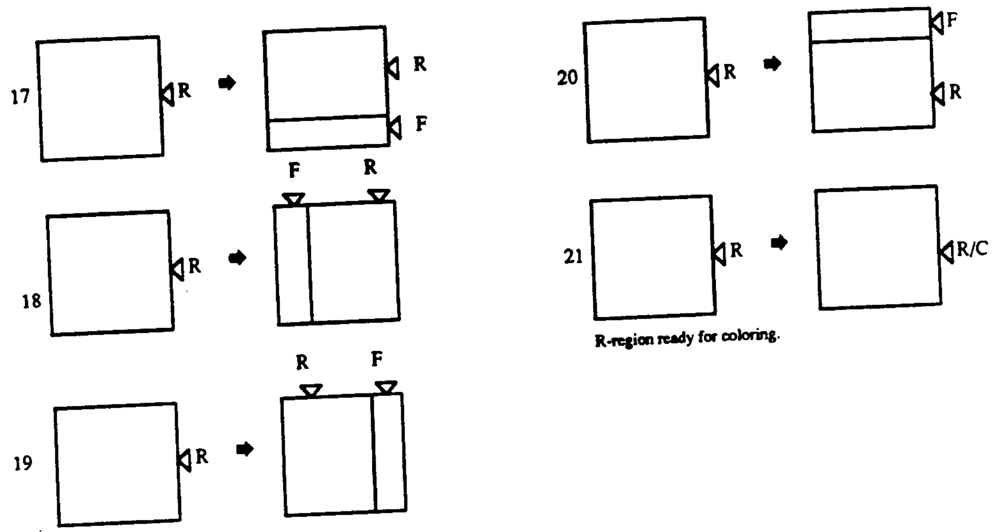


Rules for development of R-regions of the three dispather types.

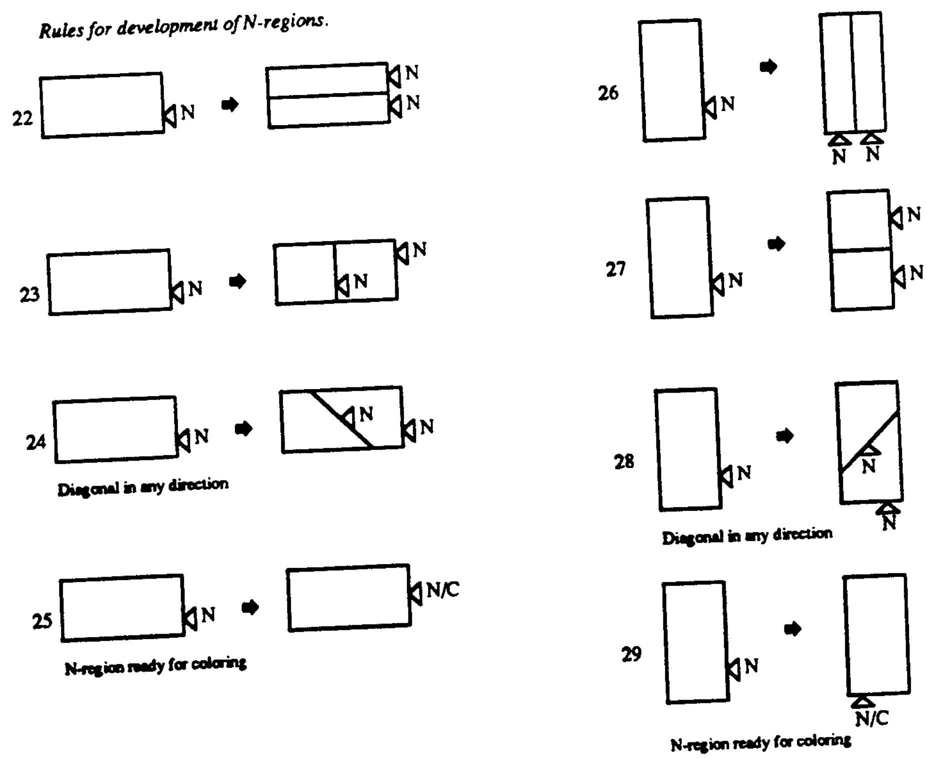




Rules for development of R-regions of unlabeled type.

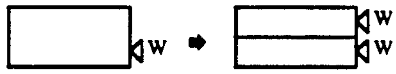


Rules for development of N-regions.



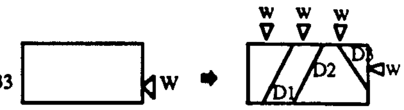
Rules for development of W-regions.

30 

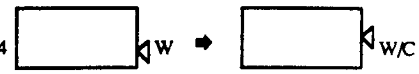
31 

32 

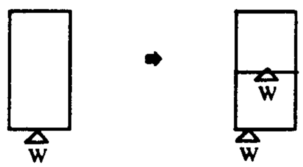
Diagonal may be drawn between any line extensions or edges.

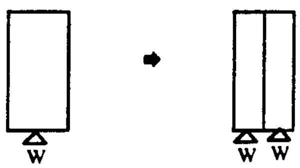
33 

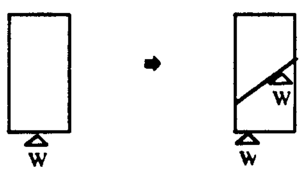
D1, D2, D3 may be drawn between any line extensions or edges. D1 and D2 are parallel within 15°. D3 is perpendicular to D1 or D2 within 30°.

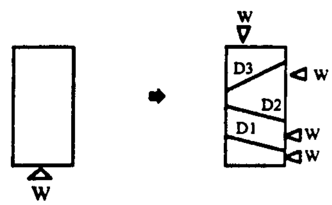
34 

W-region ready for coloring.

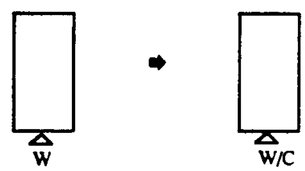
35 

36 

37 

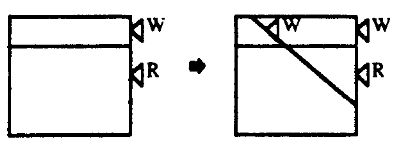
38 

D1, D2, D3 may be drawn between any line extensions or edges. D1 and D2 are parallel within 15°. D3 is perpendicular to D1 or D2 within 30°.

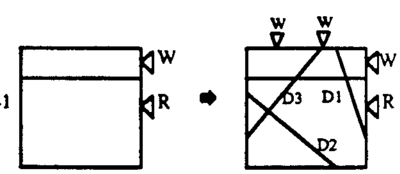
39 

W-region ready for coloring

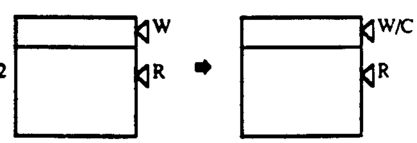
Rules for development of W and R-regions.

40 

The diagonal may be drawn between any line extensions or edges. Note that the W-region is partitioned into two W-regions, whereas the R-region is not partitioned and remains a single rectangular region.

41 

The diagonals may be drawn between any line extensions or edges. D1 and D2 are parallel within 15°. D3 is perpendicular to D1 or D2 within 30°. W is partitioned, but not R.

42 

W is ready for coloring. R may be further developed.