Environment and Planning B: Planning and Design, 1986, volume 13, pages 163-176

The structure of paintings: formal grammar and design[†]

J L Kirsch, R A Kirsch
The Sturvil Corporation, PO Box 157, Clarksburg, MD 20871-0157, USA
Received 17 January 1985

Abstract. Although substantial literature exists on the properties of formal grammars, much less has been written on the use of grammars for describing the languages which they are capable of explaining. Thus we see a well-developed theory concerning the expressive (generative) power of different kinds of formal grammars and powerful algorithmic methods (analytical) for the languages defined by these grammars. For natural spoken and written languages, grammars have been used for explaining their structure but for the two-dimensional generalization to designs, pictures, images, and fine arts, which has been known since 1964, almost no use has been made of grammars.

The purpose of this paper is to call attention to the powerful dormant tools that can be used in the design arts. Some of the benefits of using these tools for describing existing (natural) design languages are discussed. How such uses may direct us from the more formal design arts into the fine arts is discussed by reporting progress on building a grammar for a class of contemporary paintings.

1 Brief history of formal grammars

Mathematicians, logicians, and philosophers have been interested, for a long time, in making precise the notion of computation. This program has led to many independent efforts to characterize the notion of computation by defining computational mechanisms. Before the time of computers these mechanisms were, of course, formal but not mechanical in the sense that we think of them with electronic technology. Nevertheless, many such attempts were made, including, more recently, those of Turing (Turing machines), Markov (normal algorithms), Kleene (recursive functioins), and Post (production systems). Were these all independent attempts leading to differing results, the accumulated literature would be of only incidental interest. However, it has been demonstrated that, remarkably enough, each of these independent attempts converged on the same class of computational capabilities. Thus it has been demonstrated that the class of computations capable of being performed by any one or all of these computation mechanisms was precisely the same class. This led to the thesis, of Church, that the class of effective computations is this class which is defined by every one of the formal systems previously mentioned.

Church's thesis has thus led to serious study of the class of effective calculations and the now highly ramified theory of algorithms. One may thus study the theory of algorithms either abstractly, independent of any characterizing mechanism, or, by choosing one of the characterizing formal mechanisms, study the class of effective computations. We thus have, in books like that by Rogers (1967), the heavy use of Church's thesis to proceed with the study of effective computation, whereas, in books like that of Davis (1958), we have a detailed study of one particular mechanism (Turing machines) for conducting the same study.

There is, however, an independent reason for choosing one particular formal mechanism. By the 1950s, it was becoming clear to logicians that the production

† Paper presented at the NSF Workshop on the Computational Foundations of Architectural Design, held 17-19 January 1985, at the Graduate School of Architecture and Urban Planning, University of California at Los Angeles.

systems of Post (see Davis, 1958, chapter 6) could be used for making explicit the particular class of algorithms called grammars. The notion of a grammar had existed for many years in philosophy and linguistics but had always been used in a metaphorical sense. It was a suggestion by Bar-Hillel (1964), popularized and elaborated by Chomsky (1965), that led to the identification of Post's formal systems with the algorithmic notion of grammars. Linguists had been accustomed to using notions like 'immediate constituent analysis' to describe the hierarchical structure of syntactic categories used in describing grammatical structure. Chomsky's work made it possible to use Post's formal systems for making the notion of immediate constituent analysis both computationally precise and practically effective through the use of, then novel, computers.

There began a widespread investigation both of the theoretical power of such formal grammars and of empirical investigations of the use of these formal systems in describing actual languages. Some early important results included the classification of a hierarchy of formal systems by Chomsky and a demonstration of their theoretical generative power. Among the early empirical investigations was that of Yngve (1960), demonstrating the evidence for the weak generative requirements for natural languages, thereby negating some of the theoretical claims previously made. By 1964 the extent of successful use of formal grammars for describing natural spoken language and its written counterpart led Kirsch (1964) to extend the notion to handle the class of two-dimensional objects including pictures and designs. Out of this work grew the field of syntactic pattern recognition. Most of the development of syntactic pattern recognition was directed toward the problem of recognizing certain kinds of images, largely in natural photographs. However, in the past twenty years research has largely been devoted to designing tools both computational and formal for use in syntactic pattern recognition (for example, see Fu, 1982). The actual use of these tools for describing classes of images was conspicuous by its absence. A few notable examples of the use of grammars for actual designs exist, but, largely, the efforts have been devoted to elaborating the class of tools and demonstrating their use in artificial cases to study the generative and analytical power of the tools thereby designed.

2 Uses of theoretical results in grammar writing

This substantial body of theory on formal grammars is useful to the person writing practical grammars. The first class of useful theoretical results concern generative power. There is a hierarchy of formal grammars which are known to have different capabilities for generating theoretical classes of languages. Thus, for example, formal grammars, which invoke only a finite amount of memory (the so-called finite state grammars), are known to be able to produce only those languages which ultimately are periodic in their behavior. This excludes, therefore, languages which have nesting or recursion. At the other extreme, grammars which make use of the full generative power of production systems are known to be able to produce all classes of languages which can be enumerated by algorithmic methods. The person writing a grammar for a class of languages or designs is therefore confronted with a choice from a hierarchy of formal models in deciding what vehicle to use in writing the grammar.

The most tempting choice is the grammar mechanism which provides maximum generative power. Thus we have algorithms written in general programming languages which invoke maximum generative power to be able to describe existing languages. The danger inherent in this approach, however, is that such grammars may make it impossible to do analysis. It is known that there is a reciprocal relationship between the generative power of a grammar and its capability of serving as the basis for doing analysis. If one uses a finite state grammar for describing a language, then large classes of analytical questions become answerable by algorithmic means. In particular, the possibility of reconstructing the derivation of a sentence or a design from a finite state grammar is a theoretical and practical possibility whereas a sentence or design generated from a more powerful generative grammar need not necessarily be analyzable.

A grammar is thus seen as a formal statement about the structure of the language or designs to be described. Whether or not that formal structure can be used equally easily for generation and analysis is determined largely by the choice of grammar model and its position in the hierarchy of generative power. The choice of too weak a generative model prevents the description of the given language or designs. The choice of too powerful a model prevents the languages or designs thereby described from being analyzed with respect to the grammar. The optimum choice leads both to generative capability and to analyzability.

Theoretical models can also serve as new sources of insight into the structure of designs of languages. In the important early work of Yngve (1960), a grammar model was chosen for investigating English sentence structure. This model, the so-called phrase structure grammar, was used to account for the structure of English sentences. After such a grammar had been written, it was noticed that the formal power of such a model was not, in fact, used in the actual grammar. Rather, it was observed that a simpler model, the finite state grammar, was adequate for accounting for the data. This led to the important discovery of the depth restriction on natural languages. This depth restriction coincides with the observable psychological phenomenon of the limits on temporary memory in certain kinds of cognitive tasks posited by Miller (1956). The formal models provide suggestive opportunities for the creation of new insights which can then be substantiated by empirical investigations. The scenario for this class of discoveries consists of, first, the choice of a formal grammar model, then its use in empirical investigations for a class of languages or designs, followed by a study of the properties of the formal grammar thus employed, after which a simplification of the formal model sometimes presents itself. This simplification leads to a rewriting of the grammar with a less powerful formal mechanism and then suggestions, largely of a cognitive science nature, about the mechanisms used in the production of the language or design by the speakers or designers whose behavior is being described. For natural spoken languages where there exists not only the large body of theoretical results and empirical investigations in language behavior, but also the substantial body of psycholinguistic theory, this scenario offers attractive opportunities. Since much less is known about the psychology of design, the opportunities in this area are yet to be demonstrated.

3 The uses of grammars

We have already seen an instance of the use of grammars to suggest new cognitive science hypotheses. This is an example of the most important reason for using formal grammars to describe behavior, whether it be of a linguistic or of a design nature. The grammar serves as a powerful vehicle for expressing insight.

The scholar, wishing to describe a class of behaviors, finds that the use of a formal grammar is a powerful vehicle for making statements about the structure of the behavior being described. In the case of language behavior this structure is the syntactical structure of the language. In the case of design behavior this structure is the compositional structure of the design. The grammar does not, of course, provide a discovery procedure for the structure. That is the consequence of scholarly investigation and creative insight. But the formal grammar is ready to be used to express that insight in such a way that the consequences of the insight can be powerfully used.

The most obvious use of a grammar for testing insight into the structure of designs or languages is for generation. For example, in Koning and Eizenberg's (1981) grammar for the designs of Frank Lloyd Wright, the grammar serves to describe the structure of a few actual designs. But then, by using the grammar as a generator, further designs can be generated which can then be inspected for their structural similarities to the actual ones on which the grammar was based. The extent to which such additionally generated designs are convincingly realistic is the extent to which the insight in the grammar is plausible.

Use of grammars for generation is often confused with the possibility of making new designs or creating new language. New language seldom occurs, because new designs or new language produced by a grammar are structurally not innovative but only inherent in the behavior from which the grammar was originally built. Generation from a grammar is a debugging tool for use in debugging theories and for use in validating them based on plausibility criteria. If such generation can lead to additional instances of existing language behavior or designs, then those instances may be of some practical use. But, typically, they will not transcend the corpus on which the grammar was designed.

Another use of formal grammars is in parsing, that is, determining the syntactic structure of particular instances being analyzed. The reason why parsing is useful for natural language is that language utterances do not exist out of context. They are part of a large body of comparable utterances and, to a significant extent, the information or, perhaps, meaning inherent in such utterances resides in the extent to which they are distinguished from other utterances sharing much of the same syntactic structure. One expects this same notion to carry over into designs where the aspect of a design which distinguishes it structurally from others of its class is the part that conveys information or innovation. Thus to be able syntactically to analyze or to parse a design is to be able to distinguish it from other related ones, bearing much of the same structure but with particular variants that are unique to the design being analyzed. Thus, designs are information bearing and their information can be made explicit by syntactic analysis with respect to formal grammars.

Another use of grammars is in problem decomposition. Suppose that a design problem or a language problem is solved by writing a grammar for the corpus to be described. The grammar will typically consist of a set of production rules. It may be determined after the grammar is written that these rules can be divided into disjoint sets having no logical connection between sets. Each of these sets, then, constitutes a subgrammar or a decomposed part of the whole problem. Each of these subgrammars can stand on its own as a grammar for the subpart of the corpus which it describes. In writing large grammars it is possible to use subgrammars in the same sense that subroutines are used in computer programming. They may be invoked repetitively and may be interchanged among different grammars. The notion that a style by a designer or language user is characterized by the structure of use of subgrammars is an important notion in stylistic analysis. The possibility of using grammars to explicate such stylistic analysis is an important opportunity presented by the use of formal grammars.

4 Extension of formal grammars to design

Thus far, we have been implicitly assuming that the notion of grammar as it occurs in language theory can be extended to formal design and that some of the benefits of using grammars in language description can accrue to the description of designs. In this section, we will outline some of the progress previously made in using grammars for designs.

The original work in this field dates back to Kirsch (1964) who demonstrated that formal phrase structure language production systems could be used in a strictly analogous sense in generating designs. No example of a grammar for an interesting naturally occurring class of designs existed at that time. Shortly after, there began work on a number of so-called 'toy grammars' for things like printed characters, and simple geometric patterns. The first significant grammar for designs was the grammar of Nevada cattle brands by Watt (1966). He used the customary methods of structural linguistics to describe the class of geometric shapes used for denoting cattle brands and also used the same methods to describe the spoken version, the so-called blazons, that corresponded to these designs.

The natural temptation in the early days of syntactic analysis of designs was to deal with highly formalized designs. Thus, some unpublished studies were made of mathematical notation at the National Bureau of Standards to try to describe the structure of mathematical forms used in logic and the propositional calculus. Preliminary investigations showed that to account for even such a small part of mathematics as set theory, propositional calculus, and Boolean algebra, would require some twenty pages of grammar rules. An attempt was made by Rankin et al (1965) to describe the syntactic structure of Chinese characters. The Chinese characters were shown to have a compositional structure which could be described with grammar rules in very much the way that Watt had demonstrated for Nevada cattle brands. But it was some years before an actual class of designs created by a designer were accounted for by a grammar. First, Stiny and Mitchell (1978) developed a grammar to generate Palladian villas, and then Knight (1980) provided a grammar for Hepplewhite chair-back designs. And of course, there was the important result of Koning and Eizenberg (1981). They were able to demonstrate that a grammar for Frank Lloyd Wright's prairie houses could be written in some ninety-nine production rules using the shape grammar system of Stiny (1980).

Despite the sparse activity in actually writing grammars, considerable activity existed in developing the tools of syntactic analysis for generation of designs and patterns. Some of this work is surveyed in the book by Fu (1982).

Thus we find the situation in 1985, wherein highly developed tools for formal grammars of design exist but sparse use has been made of these tools. Of course, many of the tools have been built by researchers in computer science and one would expect the tools to be used by researchers in the disciplines wherein the designs occur. Thus the unusual cooccurrence of computer science and, for example, architecture is required for significant efforts in using grammars in architectural design. It is reasonable to expect, nevertheless, now that both the tools and a significant use of them in design have been demonstrated, that there will be a rapid expansion of effort in this area.

5 Design basis in the fine arts

If one may assume that the use of formal grammars in design, in such fields as architecture and graphic design, is to proliferate in the near future, it is reasonable to enquire about the next direction in which these formal tools may be used. One such candidate appears to be the use of grammars in the fine arts. There is certainly a design basis for much work in the fine arts, even though finished work usually transcends the design stage often evident in preliminary sketches or models. Even where the preliminary designs are not made explicitly by the artist, the possibility of an ex post facto explanation of the artwork, using a design stage, is an interesting possibility. This is an example of the approach taken by Loran (1943) in describing Cézanne's compositions. He used diagrams and sketches to show how Cézanne organized his paintings, without there being any necessary evidence that such sketches were, in fact, specifically planned by Cézanne. Of course, there are cases in which paintings can be analyzed to show underlying pentimenti which explicitly exhibit the preliminary design stage.

Borrowing from structural linguistic terminology, one may consider that there are a deep structure and a surface structure associated with an artwork. The surface structure accounts for many of the observable properties of the finished work. These include, for example, texture, variation of media, line quality, and colors and their relationships. The deep structure can account for the overall composition and how the work is organized in two or three dimensions. Deep structure can also account for the interesting property of recursion. Recursion occurs in a grammar when a formal production rule leads ultimately to invoking itself, thus allowing for infinite recurrences of the same structure. We will see, below, that this kind of recursive structure occurs in certain paintings. Incidentally, it may be used to account for the notion of three-dimensional depth in artworks. The formal properties of recursion and of depth are similar enough to consider that recursion in grammars might be a mechanism for making explicit the properties of depth and perspective used in organizing three-dimensional space. Extensive occurrences of recursion are present in the grammar, discussed below, for the paintings of Richard Diebenkorn.

6 A formal architectural grammar for Richard Diebenkorn's Ocean Park paintings

Richard Diebenkorn is one of the most important and respected contemporary American painters. Although he has explored several styles, we have chosen to examine one: his well-defined Ocean Park series (see Buck et al, 1980). Between 1967 and 1983, he painted about 135 very large abstract oil paintings, influenced by the luminosity, color, space, and architecture of the Ocean Park area of Santa Monica, CA where he lives and works.

We chose these works as a first attempt, in part, because they are roughly geometric, and therefore appear to be conventionally describable and measurable. Actually, they are extremely complex and varied, asymmetrical, and immediately characterized by ambiguity, and they therefore offer a challenge beyond a trivial geometric statement. A major aspect of Diebenkorn's complexity and ambiguity arises out of evidence of his drawing process. Even though one can only glean an understanding from the finished work, in Diebenkorn's case, the pentimenti not only are a vital clue to the evolution of his composition, but figure strategically in the grammar of his work and in the finished paintings.

The grammar we describe here deals primarily with the linear compositions and the resulting areas. We deal with color only summarily insofar as it defines the spaces, but not for its inherent character or effects. Nor do we treat texture or line quality at this time. Of course, questions of intention, meaning, content, and metaphor we treat not at all, these being well beyond the scope of our formal system. Our restriction to line drawing in no way implies its primacy in Diebenkorn's work. It is, rather, the exquisite integration of the many plastic qualities that defines his painting and gives it meaning beyond its parts. For the present, it is easier to isolate line for a prototype grammar.

Currently the grammar can account for deep structure, and can distinguish it from surface structure. The deep structure, or the basic composition, is indicated by lines and regions, often masked by overpainting, and has been described with a set of shape grammar and phrase structure rules. Color, texture, and line quality, so important to the whole picture, appear to be transformations, applied to the deep structure, that is, they form the surface structure.

As previously mentioned, the grammar can also account for a partial, plausible time sequence at the deep structure level. For example, at T-intersections of the linear elements, we can reasonably assume that the top of the T is drawn before the leg of the T. Such considerations allow us to infer some details of the compositional sequence. We can also assume that what is prior in time is also prior in organizational importance. Thus, high-level organizational decisions show up in the earlier part of the composition.

Diebenkorn's habit of recursive composition is particularly noticeable and is intrinsic to writing the grammar with rules that refer back to themselves. At the highest level of organization of the grammar, there are large rectangular regions which are divided into multiple smaller regions. Some of these smaller regions have the same internal structure as the parent regions. This produces the recursion. (A similar kind of recursive structure is evident in Stiny's (1977) grammar for Chinese ice-ray lattice designs.) Naturally, there is no obvious limit to how many levels of recursion are possible. Although the grammar thus allows infinite recursive nesting, no arbitrary way of limiting the recursion seems warranted. It seems preferable to allow the (absurd) infinite recursion than to limit it arbitrarily at some finite number. We might note, in passing, that this approach is similar to one followed currently in computer graphics where fractal curves are used to produce realistic scenes. The definition of the fractals allows infinite recursion, although no more than a few levels are visible in any scene.

The grammar we have devised (see the appendix) consists of a set of production rules in a form similar both to Stiny's (1980) shape grammars and to context-dependent phrase structure grammars. As in shape grammars, we use a set of labels as control structures to regulate the applicability of the production rules. These alphabetic symbols are attached by an arrow pointing to a syntactic constituent. One notational device we use is borrowed from one of the early programming languages, COMIT (Yngve, 1959). This is the so-called dispatcher (later called property lists in languages like LISP). In a rule such as OPP → OP/S the dispatcher, S, is a property added when the rule is applied and inherited in all subsequent rule applications unless specifically removed by a rule. When the dispatcher appears on the left-hand side of a rule, it serves as a condition that must be met for the rule to be applicable. This notational device can be eliminated at the expense of proliferation of the number of rules.

When lines are drawn to define constituent regions, lines ending at the edge of a grammatical constituent region may be extended to cross that region. Such lines are denoted by dotting them when such an option is exercised.

On some of the rules, we specify geometric conditions that must be met. These can be interpreted as Stiny's predicates. Again, these predicates result in a large saving in number of rules. Such predicates are used to specify angles. They may also be used for other geometric properties like region width and line length.

The formal part of the grammar accounts for the deep structure of the paintings. This deep structure is indicated in the finished paintings, by pentimenti and by borders between adjacent regions. The surface structure requires more than we provide here. Some of that surface structure is partially described by a set of 'coloring rules'. By 'color' we mean to suggest actual color, texture, and surface quality, but in this grammar, we only allow distinctions to be made between regions differently colored, without attempting to account for the actual values of these surface properties. For this reason, the black and white illustration in figure 1 of Ocean Park number 111 is an adequate illustration for testing the grammar, although it cannot hope to capture the rich variety of surface properties in the actual Diebenkorn painting.

After all formal rules have been applied, the coloring process may be applied. F-regions are colored similarly to their adjacent R-regions. The W-regions with dispatchers /S and /U are assigned different colors from adjacent W-regions; those with /R are assigned colors similar to those of adjacent W-regions. N-regions are colored arbitrarily. When a region is colored, lines that traverse the region are 'ghosted', as they would be by overpainting. They thus remain visible but are deemphasized. After coloring is complete, some ghosted lines may be reemphasized by repainting with colors distinct from those of the two or more regions bordering the line.

The two tests that may be applied to the grammar given in the appendix are analysis and synthesis. Analysis is applied to the corpus of existing paintings to determine whether compositional phenomena, used by the painter, can plausibly be furnished by the grammar. Synthesis is used to generate compositions to determine whether the grammar specifies particular compositional phenomena that cannot, plausibly, be attributed to an extension of the painter's oeuvre. We apply both kinds of tests here to enable the reader to make such plausibility tests.

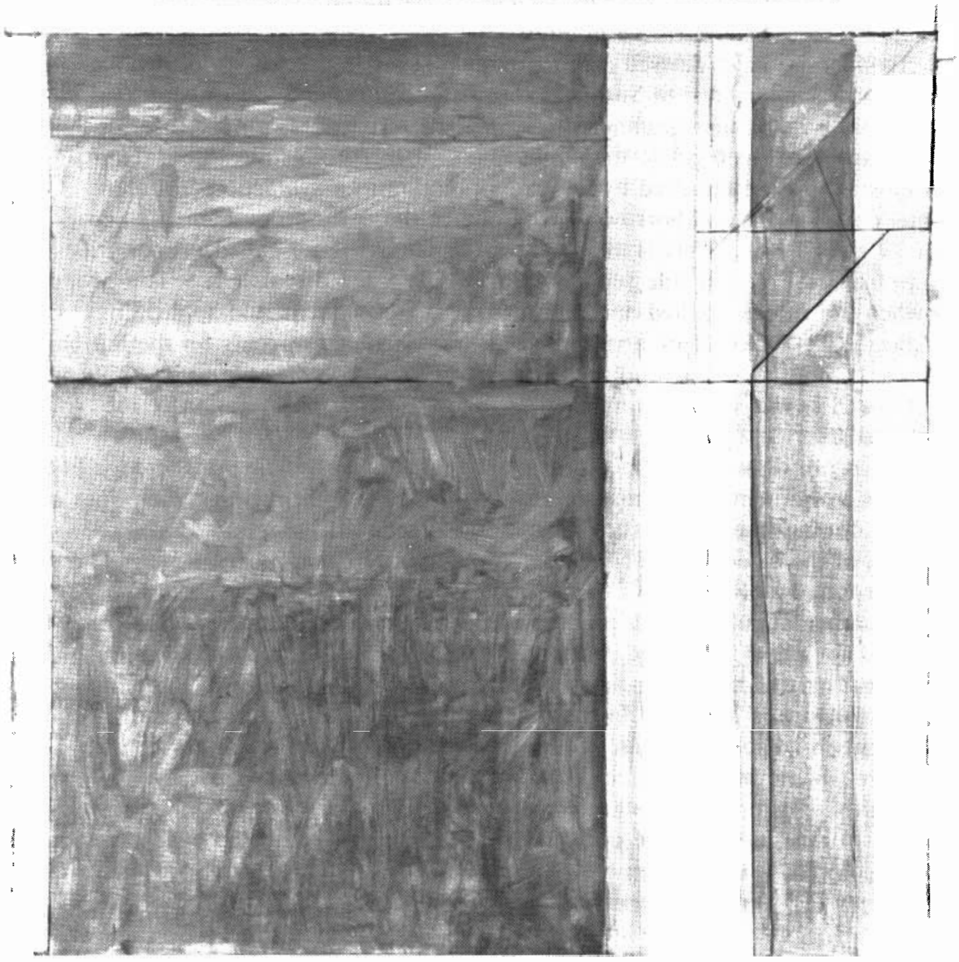
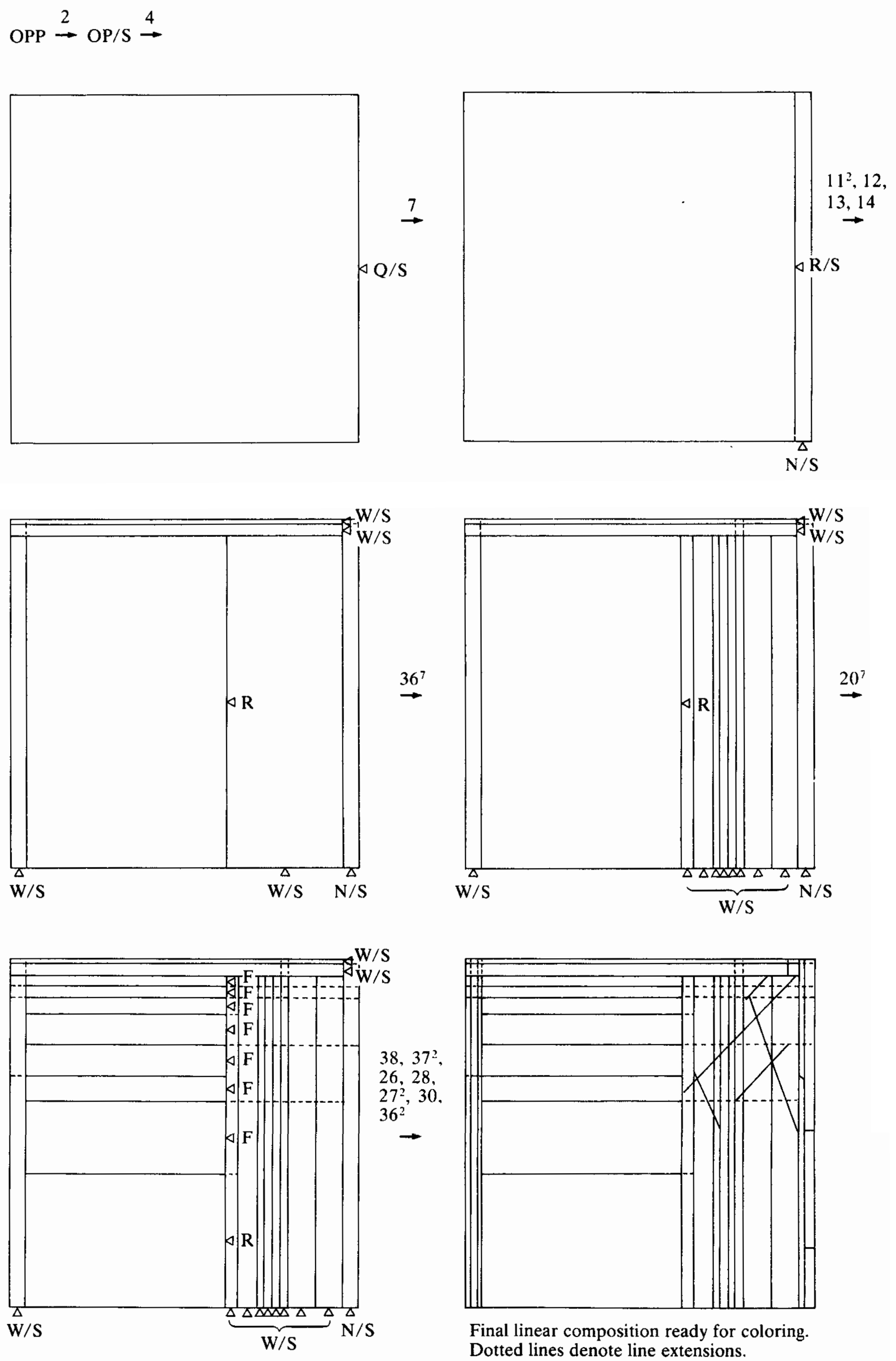


Figure 1. Ocean Park number 111 by Richard Diebenkorn, 1978, oil and charcoal on canvas, 336.2 × 336.7 cm (by courtesy of the Hirshhorn Museum and Sculpture Garden, Smithsonian Institution).



Final linear composition ready for coloring. Dotted lines denote line extensions.

Figure 2. Grammatical derivation of linear composition for Diebenkorn's Ocean Park number 111. (The superscripts on rules indicate the number of times a rule is to be applied in succession.)

For analysis, we start with Diebenkorn's Ocean Park number 111, shown in figure 1. Careful inspection, both of the original painting and of the reproduction, allows a linear compositional diagram of the painting to be drawn. It remains to determine whether the linear composition can be derived from the grammar. This is demonstrated in figure 2. Here we see grammar rule applications that may be successively applied to the starting symbol, OPP, to derive a linear composition. The target composition, the last square in figure 2, is obtained by using the thirty-three rule applications shown. The reader may compare this composition with the evidence appearing in figure 1. Similar analysis tests may be applied to the grammar for other Ocean Park paintings and their linear compositional analyses.

The synthesis test is applied by generating a linear composition randomly from the grammar. This is easily done with a computer program, or manually, as we have done to produce figure 3, which required seventeen rule applications. This linear composition does not correspond to any of the Ocean Park paintings known to us. It may be judged, by those knowledgeable with Diebenkorn's oeuvre, as to whether it is a plausible extension.

In producing figure 3 from the grammar, several decisions must be made that are not specified by the grammar. These include the choice of applicable rules provided by the grammar where alternatives are allowed. Thus, the first rule applied, number 2, results in a particular organization of space in the composition, a kind of 'suburban' landscape, that would have been different had the 'urban' or 'rural' alternatives of rules 1 or 3 been selected. Another set of decisions concerns the dimensions of regions generated, on which the grammar is silent. Predicates to govern such dimensions could be added, just as they are in rules 15, 33, 38, and 41, to govern angles. Decisions must also be made about which lines are to be extended outside their constituents, as shown by dotted lines. These extensions are not specified by the grammar. And, of course, there are color and other surface properties to be decided upon. The grammar provides a useful framework on which these additional properties may be added.

More specific habits of Richard Diebenkorn will become evident to the patient reader who can trace through the grammar, but a final note about grammar writing is in order. It has always been the case that attempts to write grammars begin with a plan to circumscribe the corpus, often by circumscribing the subject matter. There is always a surprise for the grammar writer in the wealth of structure evident in subsets of a language which appear simple. By choosing one coherent set of paintings by one artist, we have attempted such a circumscription. We expect to be surprised.

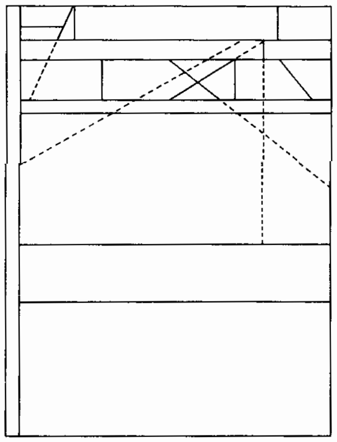


Figure 3. A pseudo-Diebenkorn derived from the following sequence of rule applications: 2, 6, 17, 17, 11, 31, 31, 31, 30, 38, 37, 30, 31, 30, 30, 30, 32.

Acknowledgements. We would like to thank George Stiny for providing the incentive for this work, both by his seminal research on shape grammars and by organizing the workshop at which this work was first presented. Kathan Brown of Crown Point Press provided her special insight into the work of Diebenkorn, as did Frank Gettings of the Hirshhorn Museum and Sculpture Garden, Smithsonian Institution. The staff of the UCLA Graduate School of Architecture and Urban Planning helped make the paper more presentable, if not more palatable, for which we thank them.

References

Bar-Hillel Y, 1964 *Language and Information* (Addison-Wesley, Reading, MA)

Buck R, Cathcart L, Nordland G, Tuchman M, 1980 *Richard Diebenkorn: Paintings and Drawings* (Albright-Knox Art Gallery; Buffalo and Rizzoli International Art Publications, New York)

Chomsky N, 1965 *Aspects of the Theory of Syntax* (MIT Press, Cambridge, MA)

Davis M, 1958 *Computability and Unsolvability* (McGraw-Hill, New York)

Fu K S, 1982 *Syntactic Pattern Recognition and Applications* (Prentice-Hall, Englewood Cliffs, NJ)

Kirsch R A, 1964, "Computer interpretation of English text and picture patterns" *IEEE Transactions on Electronic Computers* **13** 363-376

Knight T W, 1980, "The generation of Hepplewhite-style chair-back designs" *Environment and Planning B: Planning and Design* **7** 227-238

Koning H, Eizenberg J, 1981, "The language of the prairie: Frank Lloyd Wright's prairie houses" *Environment and Planning B: Planning and Design* **8** 295-323

Loran E, 1943 *Cézanne's Compositions* (University of California Press, Berkeley, CA)

Miller G A, 1956, "The magical number seven plus or minus two" *Psychological Review* **63** 81-97

Rankin B K III, Sillars W, Hsu R, 1965, "On the pictorial structure of Chinese characters" report TN254, National Bureau of Standards, Washington, DC

Rogers H Jr, 1967 *Theory of Recursive Functions and Effective Computability* (McGraw-Hill, New York)

Stiny G, 1977, "Ice-ray: a note on the generation of Chinese lattice designs" *Environment and Planning B* **4** 89-98

Stiny G, 1980, "Introduction to shape and shape grammars" *Environment and Planning B* **7** 343-351

Stiny G, Mitchell W J, 1978, "The Palladian grammar" *Environment and Planning B* **5** 5-18

Watt W C, 1966, "Morphology of the Nevada cattle brands and their blazons" report 9050 (out of print) National Bureau of Standards, Washington, DC

Yngve V H, 1959 *The COMIT Programming Language* (MIT Press, Cambridge, MA)

Yngve V H, 1960, "A model and an hypothesis for language structure" *Proceedings of the American Philosophical Society* **104** 444-466

APPENDIX

Ocean Park grammar rules

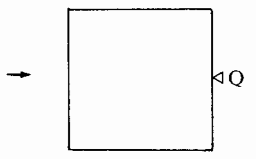
1 OPP → OP/U

2 OPP → OP/S

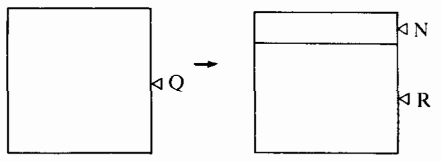
3 OPP → OP/R

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

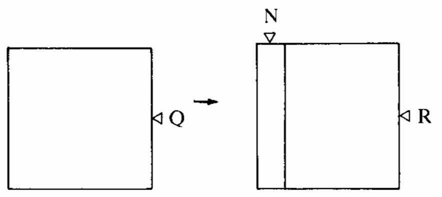
4 OP



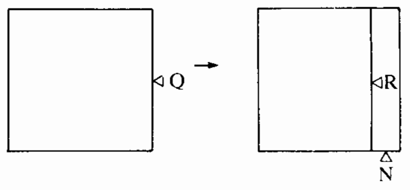
5



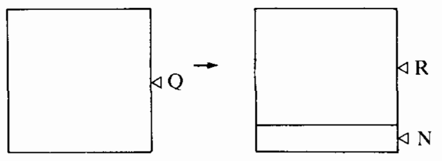
6



7

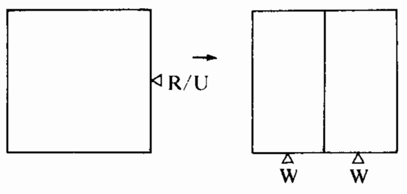


8

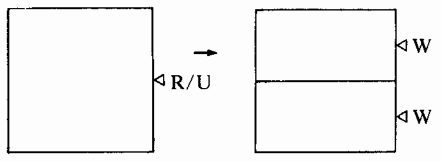


Rules for development of R-regions of the three dispatcher types

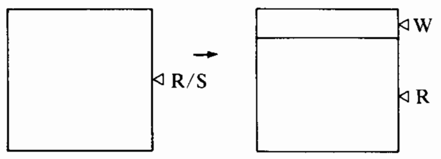
9



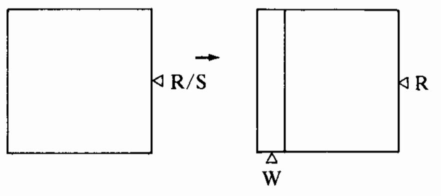
10



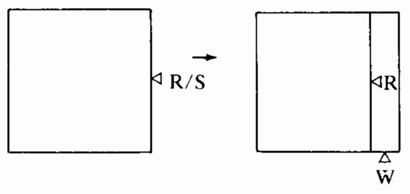
11



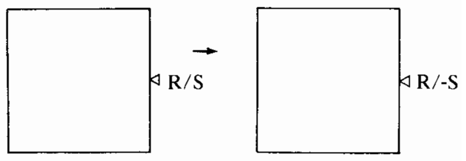
12



13

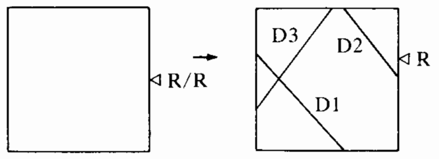


14



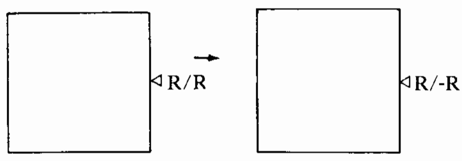
R with dispatcher S removed

15



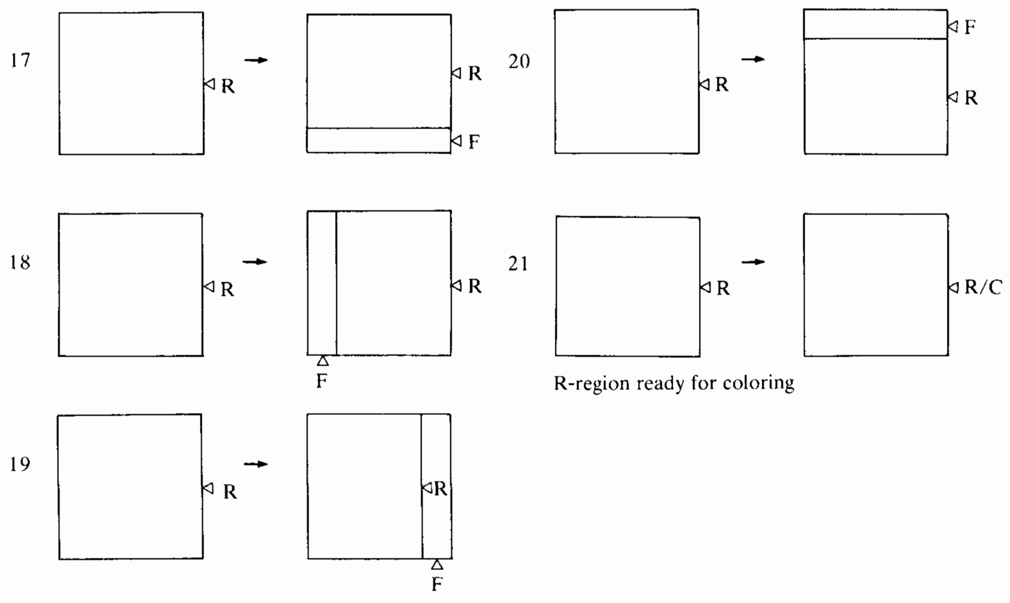
Diagonals D1, D2, D3 may be drawn between any line extensions or edges. D1 and D2 are parallel within 15°· D3 is perpendicular to D1 or D2 within 30°.

16

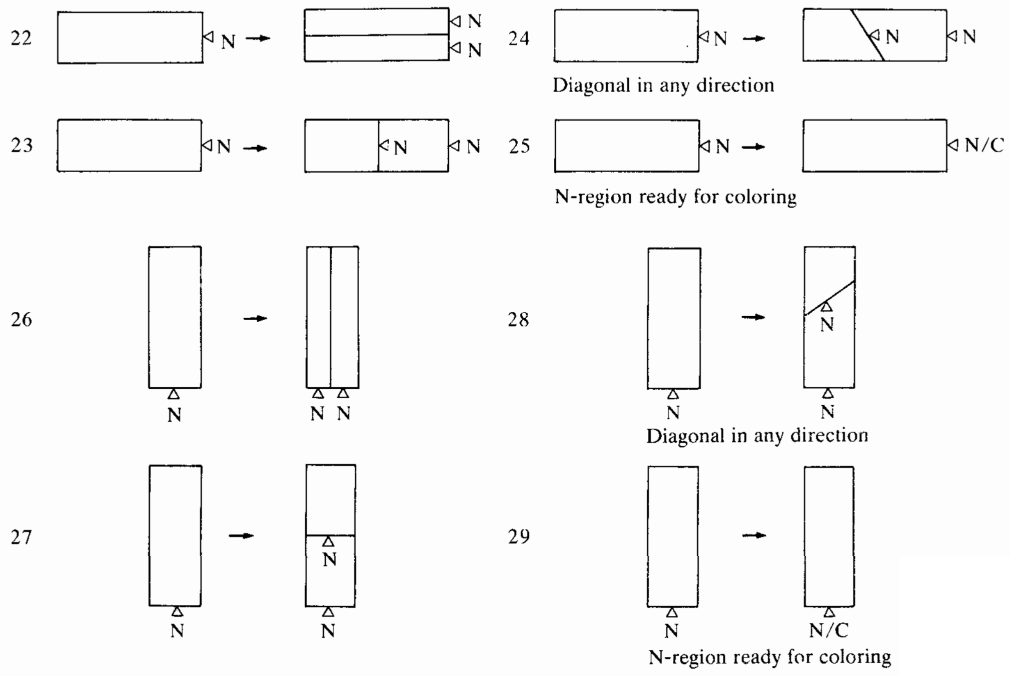


R with dispatcher R removed

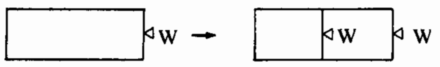
Rules for development of R-regions of unlabeled type

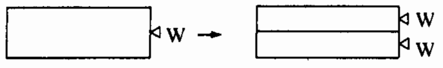


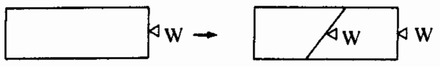
Rules for development of N-regions


22
N
N
N
23
N
N
N
24
N
N
N
Diagonal in any direction
25
N
N/C
N-region ready for coloring
26
N
N N
27
N
N
N
28
N
N
N
Diagonal in any direction
29
N
N/C
N-region ready for coloring

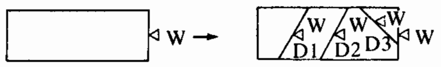
Rules for development of W-regions

30 

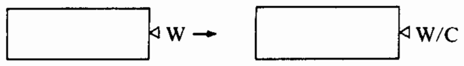
31 

32 

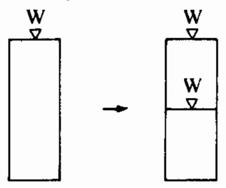
Diagonal may be drawn between any line extensions or edges

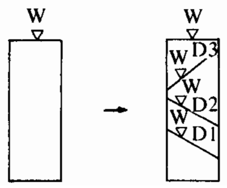
33 

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°.

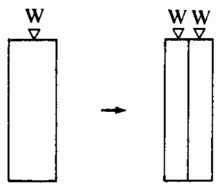
34 

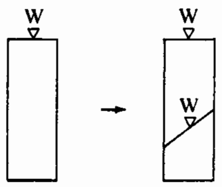
W-region ready for coloring

35 

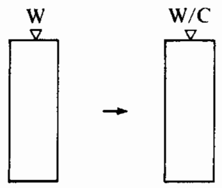
38 

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°.

36 

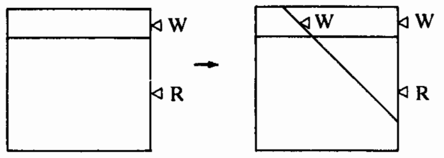
37 

Diagonal may be drawn between any line extensions or edges

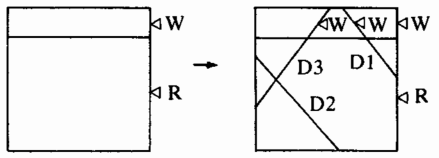
39 

W-region ready for coloring

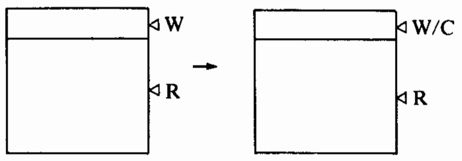
Rules for development of W- and R-regions

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 within15°. 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.