NOTICE WARNING CONCERNING COPYRIGHT RESTRICTIONS:
The copyright law of the United States (title 17, U.S. Code) governs the making of photocopies or other reproductions of copyrighted material. Any copying of this document without permission of its author may be prohibited by law.

On The Representation And Generation Of Loosely-Packed Arrangements of Rectangles

by

U. Flemming

EDRC-48-01-86 3

September 1986

On the representation and generation of loosely-packed arrangements of rectangles

Ulrich Hemming
Department of Architecture
Carnegie-Mellon University
Pittsburgh, PA 15213

26 February 1985

Abstract

Several computer programs that enumerate rectangular dissections as solutions to certain Liyoui problems have established a distinct paradigm for dealing with the crucial theoretical issues involved. The present paper suggests an extension of the paradigm to include "looscly-pjcked arrangements of rectangles', which arc of wider applicability in an architectural context. The paper introduces *orthogonal structures* to represent these arrangements and esuiblishes the conditions« r well-formedness for these structures. It presents a grammar to enumerate orthogonal structures and suggests that best use is made of the grammar if it is incorporated into a generative ex pen system, able to serve as a vehicle to discover, encode and utilize a broad range of constraints j:ui criteria in the generation of layout alternatives.

Table of Contents

1 Background	1
2 Orthogonal Structures	6
3 Generation of Orthogonal Structures	17
4 A Generative Expert System for the Design of Architectural Layouts	28

University Libraries
Carnegie Mellon University
Pittsburgh, Pennsylvania 15213

I Background

A useful classification of approaches towards the computer-assisted generation of floor plans is given in [7J:

1. Automated appraisal of layouts that have been generated by traditional means
2. Stcpwisc automatic layout generation interactively guided by manual selection of desirable partial solutions
3. Noncxhaustivc automatic generation satisfying given constraints
4. Hxhaustive automatic generation satisfying given constraints
5. Automatic generation of optimal or quasioptimal layouts under given constraints.

Among these, approach 4 demands the most elaborate theoretical foundation. It is particularly attractive for investigations in which the entire set of solutions to a given problem is to be put at the designer's (or researcher's) disposal. For example, the solution set can be systematically searched for *effluent* or *Pareto-opiimal* solutions, each of which is distinguished by a particular trade-off between lUhanta^s and disadvantages that warrant a closer analysis and comparison (see, for example. [11] for a demonstration of this situation, albeit within a different context).

Starting with [9] and [13] and continued through [10] and [5], work on the exhaustive enumeration of solution sets has produced a particular approach which, by now, has established itself as a fully developed paradigm. This paradigm achieves great conceptual clarity by drawing a clean distinction between, on the one hand, die quantitative and continuous properties of a solution (such as the dimensions of the spaces allocated) : id »n the other hand, some of its qualitative or discrete properties (particularly the geometric or spatial · ^ i necessarily topological - relations between the allocated spaces). The paradigm stresses the imp< r: tn • using a formalized representation for properties of the second type and calls for an explit spcciticat; .- • necessary and sufficient conditions under which such representations are to be considered HW/-' - • ; - *syntactically correct'*, that is, every representation of a solution satisfies these conditions and ever\ . -n v.: • • r satisfies these conditions represents a solution. In the enumeration of solution sets, these represent a: ; :. a crucial role in two ways:

(1) Each representation is an *abstraction* since it supresses certain properties of the solution it d.v s Different solutions can therefore have the same representation, and each representation con^... . describes not a single solution, but an entire class or subset of solutions. Under a suitabK : representation, the possibly infinite set of solutions is divided into a finite set of subsets whkh • • enumerated by generating all well-formed representations *as objects,* The generation itself is based * r , ,: • construction rules, and explicit proofs are required to assure that the set of well-formed represent.-.:. • s N both *closed* and *complete* under application of these rules: that is, every rule application creates a weil • • . l

representation and every such rcprcsentaion is generated by a sequence of rule applications. Apparently, **inductive** proofs of **these** results arc straight-forward if the rules arc formulated as recursive rc-writc rules.

(2) Kach representation must record the spatial relations characterizing the solutions it describes accurately **enough to allow for an explicit formulation of the** *dependent* **or** *inter-element* constraints that restrict the dimensions of the allocated spaces and vary as the the spatial relations between spaces change (an elaboration **of this point can be found in [5]). After a representation has been generated,** a **particular** member of the **subset** of **solutions described by this representation can be found** by formulating **all** constraints imposed on **the dimensions of the allocated spaces and by computing a set of dimensions which** simultaneously satisfy **these constraints. If this process fails, the subset does not contain a solution that is feasible for** die particular **design problem at hand. ITiis step can therefore be viewed as a test that determines the** *semantic correctness* **of a representation with respect to the given problem.**

This paradigm has been developed in connection with allocation problems that are restricted in two *a•. v (1) the tasks that can be solved are narrowly defined with respect to the criteria or constraints considered .mj (2) **the solutions that can be generated are limited to** ***rectangulations*** **or** ***rectangular*** *dissections* that is, **arrangements of rectangles that are 'densely-packed' within a larger rectangle. The present** paper outlines **methods for extending the applicability of the paradigm beyond both types of limitations.** The particular **directions suggested for these generalizations will be motivated through two examples.**

Example 1:

Table 1 shows the four spaces of an efficiency apartment together with dimensional and *u>p ><±*K*.il constraints commonly imposed upon the design of such apartments (it is assumed that the** area j\.i:".r;c > **bordered from the east by a corridor and from the west by an exterior wall). Figure 1 shows fou r \\» • k.:** **this problem; they were generated by the program DIS, a floor plan generator which produces :e: : , .a: dissections as solutions to design problems of the type shown (see the description of the program in |[s] | >**

The first two layouts are well-known standard solutions, while the last two, although satisfying :: . - constraints, would never be seriously considered even by inexperienced designers: they too ob lousi. t v **common principles, conventions or rules of good design. Solution 3, for example, contains a hall* a» * • - > unreasonably large and occupies valuable space along the exterior wall that could be used better ^•. -. •• . ' the other rooms. The rules violated in layouts 3 and 4 and other layouts generated by the** program >. **not explicitly stated in an architectural program or design brief, but are nevertheless used** COMM :· **designers; they might reflect years of experience, and the designers using them are often not** aware • **unless confronted with a solution that obviously violates them. For me, the most intriguing** aspect *o*\ * ***K*** • * **with the program DIS was the discovery of precisely these implicit rules of good design.** The r

Space	Dimensional constraints		Required adjacencies
1 Hallway	Min. dimension	1.20 m	East, Space 2, Space 4
2 Living/sleeping area	Min. dimension Min. area	3.60 m 22.00 m^2	West, Space 1, Space 3
3 Kitchenette	Min. dimension Min. area	1.80 m 4.20 m^2	Space 2
4 Bathroom	Min. dimension	1.80 m	Space 1

Table 1: The spaces in an efficiency apartment

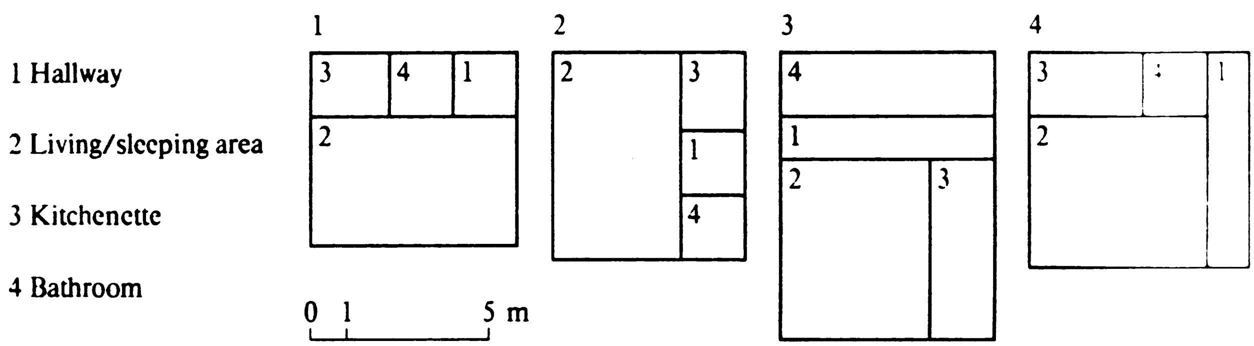

Figure 1: Four layouts satisfying the constraints of Table 1

emerged, in fact, as an effective vehicle to detect these rules, which, for the most part, are not syste... documented anywhere (e.g. in textbooks).

I was in many cases able to express these rules for a concrete task in terms of the constraints accep... program. This is, however, a laborious process and must be repeated for each new problem to be so... program would become more useful if it provided a mechanism for distinguishing between gene... rules that apply over a broad range of applications, and those constraints that specify a particular de... Rules of the first kind should be incorporated into a general knowledge base that is activated ... problem to be solved, but does not have to be explicitly specified in each case. Furthermore, the ad... new rules to the knowledge base and the modification of existing ones should be as easy as possib... involve major programming efforts. But these are precisely the characteristics of an *expert system*... means to discover and express the implicit knowledge experts use in solving problems specific to the... of expertise (see [8] and [12] for a general discussion; the latter reference contains a useful bibl...

Section 4 will specify an expert system for architectural design which, together with the generalizations suggested by **die** next example, will greatly increase the applicability of such programs as D1S.

Example2:

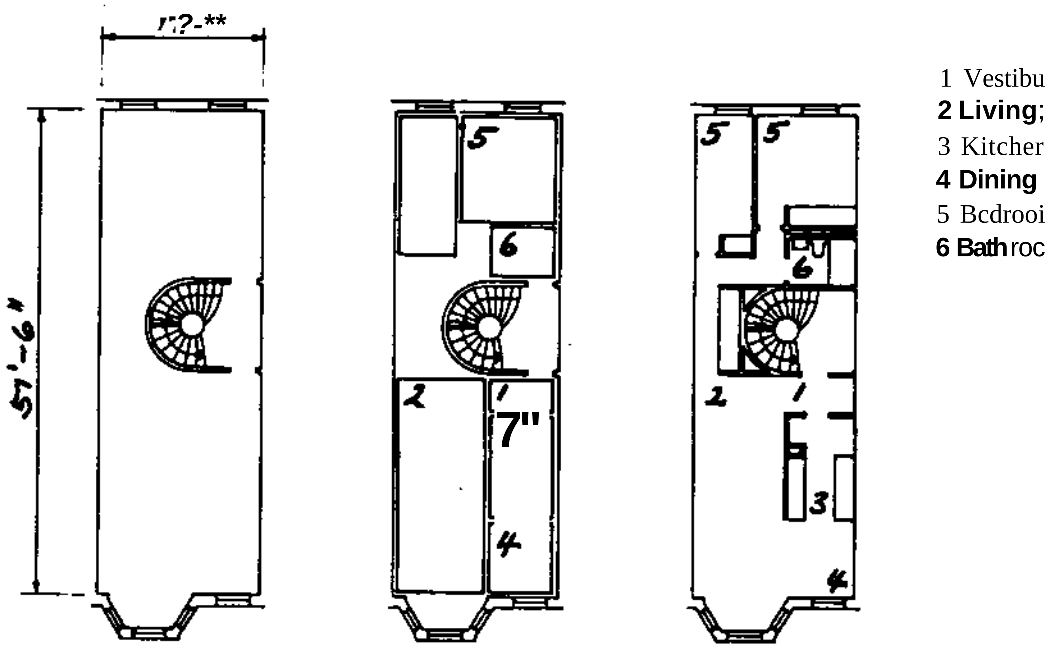

Figure 2: Stages in **the design of an apartment**

Figure 2(a) **shows the structural** walls on a typical floor of a terraced house in the Boston South f nd **once measured (while** working for John Sharratt Associates). The house was to be remodelled **jnj** into **a** multi-family dwelling with a two-bedroom apartment on each floor. Figure 2(b) shows an lnicr stage in the design process in which the major spaces have been allocated and given a rough shape stage, no attention is paid to the form of the partitions needed to separate the spaces from each mher the required circulation area delineated in any precise form. The spaces are treated more or less a s ; and prime attention is given to the relations between them (and the context).

Space-defining elements such us walls or partitions are introduced in Figure 2(c) whore the focus has shifted from spaces to the physical elements defining them. The shape of some spaces is modified in the process, and auxiliary spaces such as hallways or closets arc added.

Programs such as DIS arc inadequate to model the more strategic phase in this process. They generate densely-packed arrangements of spaces directly from a problem specification; all spaces arc immediately given a precise form and treated formally the same. Auxiliary areas must be specified at the outset along with the rooms they serve, and spaces that may have a non-rectangular outline must be divided into several components. This makes it impossible to model the staged process illustrated in Figure 2. In this process, an intermediate solution is generated in the form of a loosely-packed' arrangement of rectangles describing crucial spatial relations between the primary elements that are to be allocated; the arrangement contains gaps or holes that arc used later to allocate auxiliary spaces or that arc added to previously allocated spaces once the shape of the circulation area has been determined.

Applications of this type suggest an expansion of the paradigm to include the generation of looscly-packed arrangements of rectangles. In this expanded form, the paradigm could also be applied to the layout of equipment and furniture and similar configurations that are by definition loosely-packed.

Up to now, the most important generalization of the paradigm has been described in [4], where various structures for representing the incidence relations between the line segments and faces of *connected rectilinear shapes* (among which the rectangular dissections form a proper subset) are presented. For the applications described above, however, connectivity has little importance; primary focus is on spaces (or on the areas occupied by the objects to be allocated) rather than the lines (or walls) that separate them. The following sections indicate a distinct second direction for generalizing the results obtained for rectangular dissections.

2 Orthogonal Structures

The rectangles to be dealt with in die following arc always assumed to have sides parallel to the axes of an orthogonal system of Cartesian coordinates with a horizontal x- and a vertical y-axis. Any rectangle, r. is then completely described by the coordinates of its lower left corner, (x_z, y_z)% and by the coordinates of its upper right corner, (X_z, Y_z). where obviously

$$x_z < X_z i\backslash n d y_z < Y_z. \qquad \text{d)}$$

The spatial relations *above, below, to the left* and *to the right* arc defined on the set of rectangles as follows: If *c* and z are two rectangles,

$$c \wedge z \text{ (read } c \textit{ is above } z) <=> y_c \geq Y_z \qquad (2)$$

$$z \wedge c \text{ (read } z \textit{ is below } c) <=> c \text{ fv } z \qquad (3)$$

$$c -> z \text{(read } c \textit{ is to the left of } z) <=> X_c \leq x_z \qquad (4)$$

$$z <- c \text{(read } z \textit{ is to the right of } c) <=> c -> z. \qquad (5)$$

Obviously, each of these relations is non-symmetric, non-reflexive and transitive, *c* and z *do not overlap* if at least one of tine relations (2) to (5) holds between them.

Suppose $z\backslash . . . . z_n$ are *n* rectangles no two of which overlap. The *enclosing rectangle,* Z, is the minimum rectangle containing every rectangle z_x?, /= 1,...,/?. Z always exists and is uniquely determined. In the following, its upper, lower, left-hand and right-hand sides are always assumed to be bordered by four *e\icn<>r* rectangles labelled, respectively, N, S, W and E as shown in Figure 3. In contrast, the rectangles *z,* arc called *interior.* A set of *n* interior and four exterior rectangles, *L^* is called a *loosely-packed arrangement* r *rectangles.* Figure 3 shows the interior and exterior rectangles of a loosely-packed arrangement of recedes. L_4.

Loosely-packed arrangements of rectangles can describe various types of layouts: building parts on a MIC spaces or rooms on a floor; and equipment or furniture in a room. These arrangements are subject to v.lnous dependent and independent constraints restricting the shape of the objects to be allocated and their reUions to each other and the surrounding context. The dependent constraints can be formulated (for example. through a system of simultaneous equations and inequalities in the corner coordinates of the rcctandes) *provided that for each pair of rectangles, at least one of the spatial relations* (2) *to* (5) *has been defined.* I h.s observation suggests an expansion of the present paradigm based on the the spatial relations defined aNue.

These spatial relations cannot be selected independent of each other. For example, if *a b* and *c* arc three

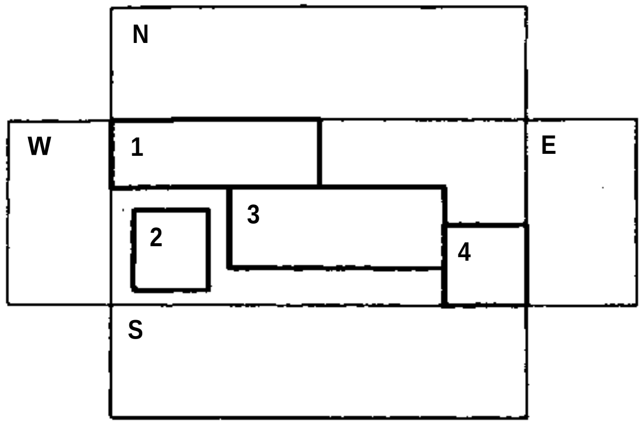

Figure 3: A loosely packed arrangement of rectangles L_4

rectangles so that a -> b and b -> c. then c -> a is impossible. In order to select, for /; given rectangles spjiul relations that can be *simultaneously realized,* a simple directed graph, (/„, is used. Its vertex set contains exactly // *interior* vertices and four *exterior* vertices labelled *N, S,* Wand *E.* Each arrow of G_n is colored in one of two colors, *h* and *w* called *horizontal* and *vertical* respectively.

The following terminology and notation are useful for subsequent developments. A path in *(i$_n$* is called *horizontal'iff every* arrow on the path is horizontal; a *vertical path* is defined in an analoguous way. It *u* , ,md *w* are three vertices so that *u* and v are connected by a directed path, />i, and v and *w* are connected *b* .i directed path, *p*± then p_x and *p*^ *have the same direction* iff v is either the starting vertex or die termin.il vertex of both *p* \ and *p* ?. For two vertices, v and w, of G_n, %

v A w (read v /5 *directly above* w) < = > G_n contains a vertical arrow pointing from *w* to v

v B *w* (read v is *directly below w)*<-> G_n contains a vertical arrow pointing from v to *w*

vLw (read v *is directly to the left ofw)* < = > G_n contains a horizontal arrow pointing from v to K

vRw (read v *is directly to the right ofw)*< = > $G_n$contains a horizontal arrow pointing from K to

Furthermore,

va w(read v *isabove* »v)< = > v A *WOT*
$G_n$contains vertices u_o(= *v*). u_l . . . .u_m(= *w)* so that for '/= 1 . . . .m, $u_{imm\,x}$ A u_f.

v b w(read v *is below w)* < = > v B *WOT*
(7„contains vertices *u$(=v),ui.* . . . .$u_m$(= *w)* so that for /= 1, . . . , *m*^*u*^*i* Bw,-.

v 1 w(read v *is to the left ofw)*< = > *v* L *WOT*

G_n contains vortices $u_o(= v),u$ \ . . . .w$_m$(= *) so that for'/= 1 . . . .///, u_{imm}\ L u_r

v r *v(rcad v *is to the right ofw)* < = > *v* K *WOT*
G_n contains vertices w$_o$(= *v).u*\ . . . .w$_w$(= w) so Lh;it for /=1 . . . .*mu$_{imm}$*\ R w,-.

For any vertex vof <?„ a(\'), /?(»»), \(v) and p(v) dcnotc the number of vertices that are, respectively, directly above, directly below, directly to the left and directly to die right of v.

The graph G_n is an *orthogonal structure* iff it satisfies the following conditions:

For every pair of distinct vertices, *v* and *w* (given in that order).
either va *w%* or vb H>, or vl *WOT* vr *w.* (5)

If vBiv, the arrow pointing from v to *w* is the only directed vertical path from v to *w:*

and if vLw, the arrow pointing from v to *w* is the only directed horizontal path from v to *w.* (7)

For every interior vertex, v\ *va* i\ v b A, v r *W* and vI*E.* (8)

SBW, SBE, NAW and *N\E.* (9)

The alternatives in condition (6) arc, as stated, exclusive; that is, any two vertices in G_n are connected b> a uniformly colored path whose direction and coloring are fixed for these two vertices.

An orthogonal structure, $G_{n\%}$ *represents* a loosely packed arrangement of rectangles, *L^* iff there OMS *A* one-to-one correspondence, / between the vertices of G_n and the rectangles of L_n mapping vertices \ / s and *W*, respectively, on rectangles N, E. S and W so that for any two vertices, v and w, of G_n

$$vLw \Rightarrow f(v) \rightarrow f(w) \text{ and } vAw \Rightarrow f(v) \uparrow f(w). \qquad (10)$$

Figure 4 shows, as an example, an orthogonal structure, G_4, and a loosely-packed arrangement ot [illegible] Z4, represented by G4.

Theorem 1: Every loosely-packed arrangement of rectangles, L_n , is represented by an orth[illegible] structure, *G^*

Proof: Any hole in L_n can be filled by additional rectangles none of which overlaps *n[illegible] rectangle in L_n or an added rectangle. The result is a rectangular dissection, L_n', with *n'i* [illegible] interior components. L_n' can be treated formally as a T-plan or trivalent dissection [5]. From / construct a directed, arrow-clored graph, G_n'. as follows. G_n' contains *n'* vertices correspond[illegible] to the interior rectangles of L_n' and four exterior vertices labelled M 5. JFand fcorrespc»nj[illegible] respectively, to the exterior rectangles N, S, W and E. G_n* contains a vertical arrow pointing *r.* [illegible] v to *w* iff the components corresponding to v and *w* border a horizontal wall or maximal line i: [illegible] the left and right, respectively. G_n> contains a horizontal arrow pointing from v^7 to *w'* iff [illegible]

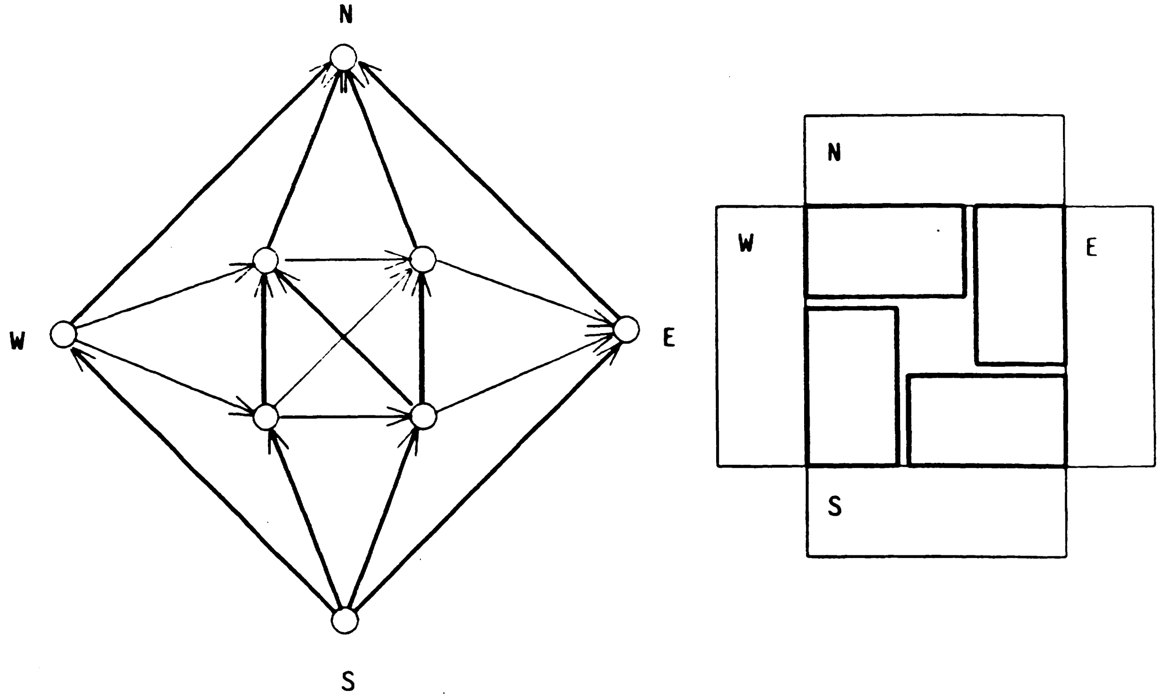

Figure 4: An orthogonal structure, G_4, and a loosely packed arrangement of rectangles, L_4, realizing G_4

components corresponding to v' and w' border a vertical maximal line from the left and right, respectively. Arrows are defined between pairs of exterior vertices according to (9). It follows from the Structure Theorem proved in [6] that the resulting graph, $G_{n'}$, satisfies (6) to (9).

Let now v be a vertex corresponding to a component not in L_n. For every pair of vertices a and b so that $a\mathbf{A}v$ and $b\mathbf{B}v$, insert a vertical arrow pointing from b to a iff a and b are not on a directed vertical path that avoids v. For every pair of vertices l and r so that $l\mathbf{L}v$ and $r\mathbf{R}v$, insert a vertical arrow pointing from l to r iff l and r are not on a directed horizontal path that avoids v. Remove v and all arrows incident with it. The resulting graph, $G_{n'-1}$, is an orthogonal structure. Repeating this reduction $n'-n$ times generates an orthogonal structure, G_n, representing L_n.

A trivalent dissection with exactly n interior components to which four suitable exterior components have been added is a special case of a loosely-packed arrangement of rectangles and will be denoted by [illegible]. [illegible] orthogonal structure constructed from such a dissection according to the process used in the proof of Theorem 1 is said to be *defined* by that dissection.

Theorem 2: Every orthogonal structure, G_n, represents a loosely packed arrangement of rectangles L_n.

Proof: By (8), every interior vertex, v, is on a directed vertical path from S to v. Define y_v as the length of the longest of these paths and $Y_v \equiv y_v + 1$. Similarly, v is on a directed horizontal path from W to v. Define x_v as the length of the longest of these paths and $X_v \equiv x_v + 1$. This gives the coordinates of n rectangles corresponding to the n interior vertices of G_n. Because of (6), no two these rectangles overlap. Adding suitably selected exterior rccuinglcs generates a loosely-packed arrangement of rectangles represented by $(i_n$.

These theorems show that orthogonal structures arc well-formed or syntactically correct representations of loosely-packed arrangements of rectangles. They can form the basis for a generator which determines, according to the current paradigm, various sets of realizable spatial relations between pairs of rectangles. If used in this way, orthogonal structures have intuitive appeal to me mainly for two reasons. They demonstrate, first of all die possibility of finding a useful structure in non-connected arrangements which do not appear amenable, at least at first sight, to the approach that has successfully been applied to connected shapes. I also find the conditions that determine the well-formedness of these structursc particularly easy to understand.

However, a note of caution must be added here. The orthogonal structure representing a loosely-packed arrangement of rectangles is not necessarily uniquely determined because certain pairs of spatial relations can hold simultaneously between two rectangles, while an orthogonal structure records only one of these relations. This problem can theoretically be resolved in two ways: (a) rules can be established under which a uniquely determined *canonical* representation is selected for any arrangement; or (b) orthogonal structures can be *refined* so that they become able to distinguish cases in which only one relation holds between two rectangles from those in which two relations hold.

None of these approaches is pursued here. For the experience with the densely-packed case, in which an analoguous problem occurs, suggests that the practical implications of this problem are negligible: as a result. the theoretical and computational complications resulting from approaches (a) or (b) become deudedK unattractive. The final judgement with respect to this situation must, however, be suspended until more experience has been gained with the present approach.

In order to develop an efficient generator, a closer look at the implications of conditions (6) to (9) is in . ' ocr

Lemma 3: Let w, v and wbe three vertices in an orthogonal structure, $<7_n$, so that *u* and v arc on i directed horizontal path, $p\backslash_{\%}$ and *w* and v are on a directed vertical path, *pi.* Then *u* and * arc either on a directed horizontal path whose direction is the same as for $p\backslash$ or on a directed \crticai path whose direction is the same as for p^.

Proof: Suppose *vxu* and va >v(see Figure 5). By (6), *w)f* v, and therefore *wfu.* Similarly, *u)6* \. a: therefore *u* JB *w.* Thus, either *w* r *u* or *u* a w. The other cases indicated in Figure 5 can be pro* analoguously.

Clearly, orthogonal structures are non-planar in the general case. They thus do not possess one important attributes shared by the structures used in [4] to represent various properties of con

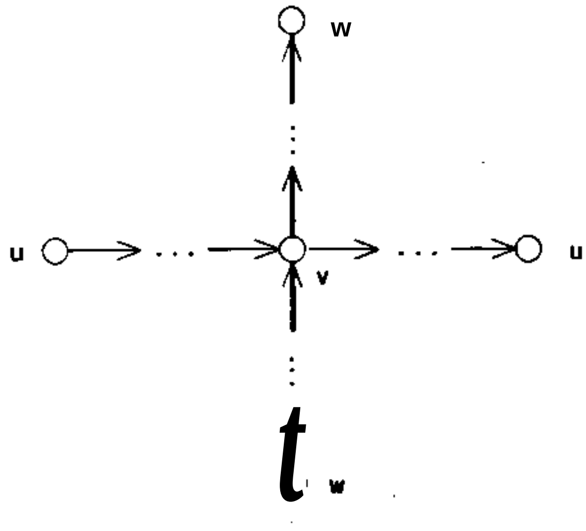

Figure 5: I .cmma 3 - Illustration

rectilinear shapes. Ilie following lemma shows, however, that orthogonal structures impl>. JI lost, an ordering on the arrows incident with a vertex.

Lemma 4: Let vbc a vertex in an orthogonal structure.

(i) If a(v)>l. the vertices directly above v form a sequence $a_{vi}, \ldots, a_{v\alpha(v)}$ so thai r..r $h=1,\ldots,\alpha(v)-1$, $a_{v,h}$ L $a_{v,h+1}$.

(ii) If/?(v)>l, the vertices directly below v form a sequence $b_{v1}, \ldots, b_{v\beta(v)}$ so thai vr $i=1,\ldots,\beta(v)-1$ $b_{v,i}$ L $b_{v,i+1}$.

(iii) If X(v)> 1, the vertices directly to the left of v form a sequence $l_{v1}, \ldots, l_{v\lambda(v)}$ so Lh.ii ' - 7=1 . . . A(v)-W V j B/,,;+ !.

(iv) If p(v)> 1, the vertices directly to the right of v form a sequence $r_{v1}, \ldots, r_{v\rho(v)}$ so ∧ i: *=1 . . . P(v)-l $r_{v,k}$ B $r_{v,k+1}$.

Proof: (i). Suppose a(v) > 1, and let a and a^x be any two distinct vertices directly above \. It. • a^7∧ a and a^7 JB a. Thus, either a' x a or a' I a. The vertices directly above v therefore i'. •• sequence $a_{v1}, \ldots, a_{v\alpha(v)}$ so that $a_{v,h}$ I ay^-j-i, /i=l . . . a(v)—1.

Suppose that for some K l≤h<a(v), a_v /, JL a_{Vth}+ \. Then there exists a vertex zso thai J r and z I fl_{Vf}^+i. z / v since otherwise, $J_{V,A}$ r v. By Lemma 3 then, z a v. But r / consequently, there exists a vertex $a_{v,h'}$ so that a_v ,//bz. lfh'<h, z\ a_v ,/,orza ay/, (by Lemir ; which is impossible. If /z^x> A+ 1, z r $<z_v$,/,+j or z a fl_{Vf}/,-i-i, which is again impossible I>. .> cannot exist, and a_{Vf}/,L$a_{V)/l+1}$.

Statements (ii), (iii) and (iv) can be proved analoguously.

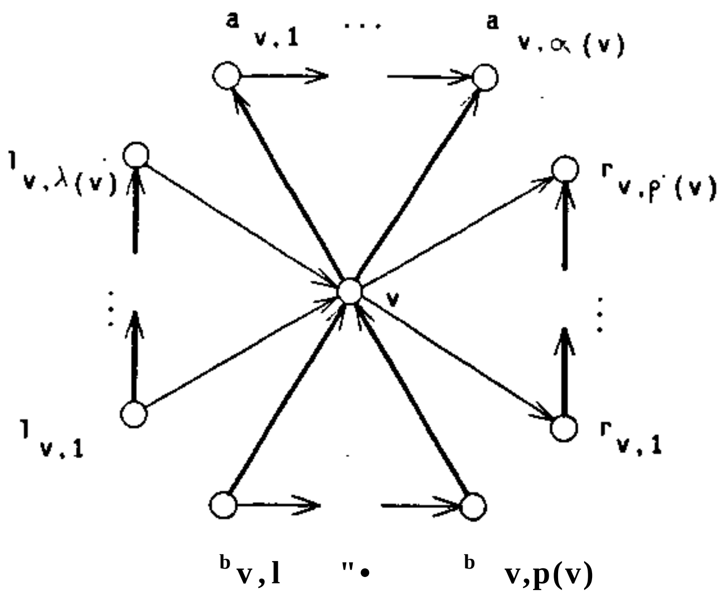

Figure 6: Lemma 4 - Illustration

The notation introduced in Lemma 4 and illustrated in Figure 6 will be used repeatedly in Lhis jnd the following section; that is, if v is a vertex in an orthogonal structure, $a_{Vi A}$ and $tf_{Vftt}(v)$ denote, respeai-. c '. Lhc first and last vertex directly above v: $/_{V\{}$ and /,,x<v) denote, respectively, the first and last vertex diivi: i. :• 'he left of v; $b_{Vt}i$ and $b_{v,}\wedge_{y)}$ denote, respectively, the first and last vertex directly below v; and final!. ' md $r_{Y,p(v)}$ denote, respectively, the first and last vertex directly to the right of v.

Lemma 5: Let v be a vertex in an orthogonal structure.

(0 A; A(v)^{B} ^{a}v, l ^{or} ^{a}v, l ^{R} ^v, A(v)-

(") ^{r}v,p(v) B ^v,$_{a}$(v) ^{or} ^{a}v,a(v) L >Vp(v)-

(iii)/$_{Vf}$l A 6$_{Vfl}$ o r 6$_{vl}$ R /$_{Yl}$.

(iv) r$_{Vi}$ i A 6$_{V|}$0(v)^{or} ^{b}v,p(v)^{L} ^{r}v,p(v)-

Proof: (i). By Lemma 3, either /$_{V}$,X(v) b tf$_{v}$,i or a_{v}\ r l_{Y}\(vy Suppose /$_{v}$,x(v) ^ ^{J}» /$_{v,}$X(y)^ a_{Vf}\. Then there exists a vertex z so that z a /$_{v?}$x(v) ^{an}d ^{z} B <3$_{M}$ i- z/ v by Lemmj - Lemma 4 then, z 1 v. But then z 1 /$_{y/}$ for some /(<X(v)). By Lemma 3, either z b I, $_{K}$ zI /$_{v,}$x(v)^ which is impossible, ztherefore cannot exist, and /$_{y}$,x(v) ^^{a}v,l-

The case where $a_{v,}i$ r /$_{v,}$ x(v) $^{as\ we}$U ^{as} statements (ii) to (iv) can be proved in an analoguous ^ 1, Lemma 6: Let vbc a vertex in an orthogonal structure.

(i) If $a_{v,1}$ R $l_{v,\lambda(v)}$ and tf$_{vl}$ r'v.l]'v.A(v)Bt;v.l and $l_{v,Mv)}$ha$v,a(v)l$

then wLv<=>ivLa$_{ViI}$and $v=b_{ayJfl}$ (H\\K => H A/$_{V>A(v)}$and r = r$_{/v.A(v),P}$</$_{v,X(v))}$].

(ii) If r$_{y\cdot p(l)}$ R $a_{v,a(v)}$ and r$_{y\ p(v)}$ b.<7$_v$, k,$_{a(v)}$ .I r$_{Vp(v)}$and $a_{va(v)}$ I r$_v$,],

then $w\backslash v<=>w\backslash r_{Ypiv)}dn<i\ \ v = l_{ry\ p(v)}Mr_{v>p(v)}$)

[w Rv<=> wR$a_{v,\alpha(v)}$ and $v= b_{a_{v,\alpha(v)},\beta(a_{v,\alpha(v)})}$].

(iii) If $b_{\%\%p(v)}$ L r$_{Vtl}$ and 6$_{Vf}$0<$_v$) I $r_{VtPiv)}[r_{vA}$ A 6$_v$,/j($_v$)and r$_{v,l}$ a 6$_{V|1}$]

then »vRv<=>ivR6$_Y$^$_v$)and v=^$_{(v)1}$a(6$_{VWv}$))I^Bv<=>wBr$_{Vfl}$_and v = $l_{r_{v,1},1}$].

(iv) If /$_{v§}$ j A 6$_{V?}$! and /$_{Vf\{}$ a i$_{vj?(v)}$ [6$_{V|x}$ R /$_{Vf}$, and $b_{v\%x}$ r /$_{v,X(v)}$],

then wBv<=> wB/$_{Vi}$i and v = r/$_{v,1}$j [H Lv<=>»vLi$_v$ j and v=a^$_{v,1}$]].

Proof: Case (i). Let /$_{Vi}$\($_V$) B a_v\ and /$_r$,x(v) I* $^a v,a(v)$- Suppose there exists a vertex : M : 'U za /$_{v<}$x($_V$) and z b $a_{v,aiv)}$. z/vsince otherwise, zI /$_v$,\($_V$) or z b /$_{Vf}$x(v)- By Lemma 3 then, : a i Hut then there exists a horizontal path from v to $ci_{Y\ a(v)}$ longer than one, which is imposM^c Tlierefore, J$_{V,}$ ^$_V$)A/$_{Vf}$x($_V$)- Furthermore, u A V=>HA/$_v$X(v) and consequently, $^{v=r}l_{\backslash iy),}p(i\backslash$ V(•

Suppose there exists a vertex z' so that z$^/$A/$_v$x($_V$) but z^x/v. z^7/v and there exists a dire^.J vertical path, which is longer than one, from v to z' and consequently from /$_v$,x($_V$) t° z'- Since :r :s is impossible, H>AV<= $W\backslash I_{v,}$ $\lambda(v)$.

The other cases can be proved analoguously. %

The four cases distinguished in Lemma 6 represent the spatial relations between a component corresponding to v and various other components that border a maximal line in a rectangular dissection. The corresponding configurations of components and lines are shown in Figure 7 (in this and some of the following figures the rectangle corresponding to a vertex is identified, for the sake of simplicity, by the label of that *verkx** T W iS observation motivates the following definition.

An orthogonal structure, $G_{n\%}$ is *dense* iff for every vertex v in $G_{n\%}$ vertices a_v\ and /$_v$-\($_V$) satisfy the prtnix. <?£ case(i), vertices tf$_v$,a(v)^{and} $^r v,p\{v)^{SdL}$^^ ^^{c} premise of case (ii), vertices r$_{vtl}$ and 6$_v$^$_v$) satisfy the premise of case (iii) and vertices $b_{Vi}\ i$ and /$_Y$ i($_V$) satisfy the premise of case (vi) in Lemma 6.

Lemma 7: An orthogonal structure, G_n< is dense iff it is defined by a dissection D_n.

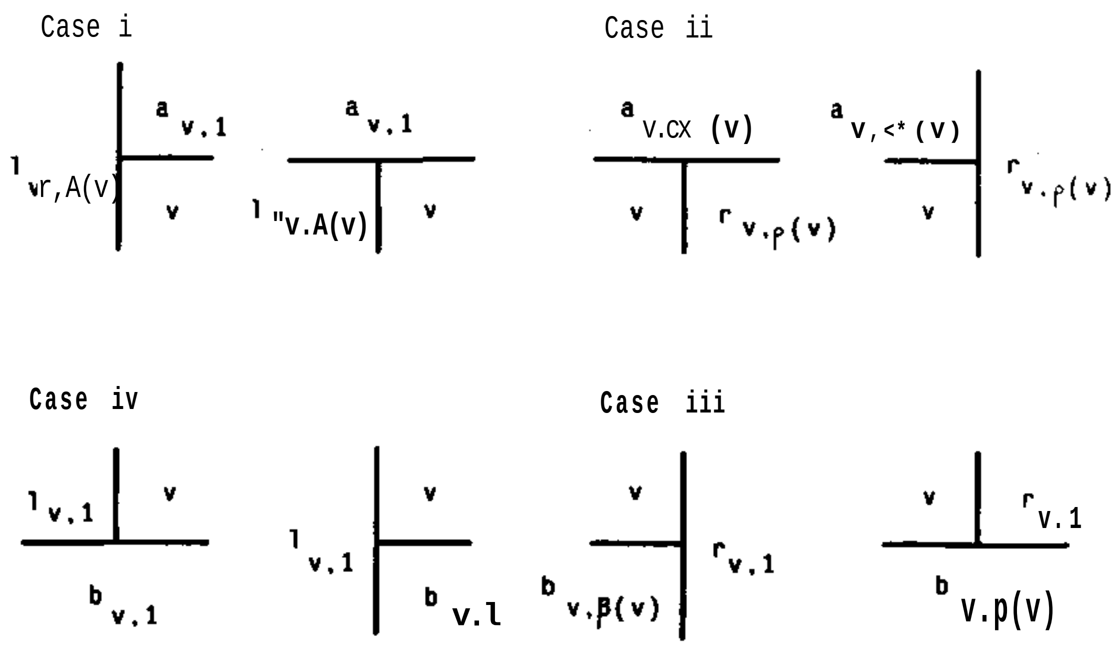

Figure 7: Lemma 6 - Illustration

Proof: *Necessity.* If G_n is defined by a dissection D_n it is obviously dense.

Sufficiency. The proof is by induction on n using operations 'in the upper left corner' as kno^n from the literature. For $n = 1$, there exists only one orthogonal structure, G_1, which is defined hylic rectangular dissection, D_1, and dense. Suppose that for some $n(\geq 1)$, every dense orihogi>rui structure with $l(\leq n)$ interior venices is defined by a trivalent rectangular dissection D_h Let (/,, <. be an arbitrary dense orthogonal structure with $n+1$ interior vertices. Clearly, $r_{w\,9}$^\ \ denote this vertex by c. Since $n \geq 1$, $b_{c\beta(c)}$ or r_{c1} must be interior, and since 6,,+ j is dense Jv vertices directly below c are precisely the vertices directly below $r_{c,1}$ or the vertices directK u» ^^ right of c are precisely the vertices directly to the right of $r_{c,1}$. The resulting two cases are sho* n •• Figure 8. Contracting, in case 1, the arrow pointing from *Wio* c or, in case 2, the arrow p(»mi;-i: from c to N creates a dense orthogonal structure, G_n, as shown in Figure 8. By hypothesis, CJL h • these structures is defined by a dissection D_n. The configurations at the upper left corner : '.' shown for each case in Figure 9. Adding a component at the upper left corner as shown m ire same figure generates a dissection D_{n+1} defining G_{n+1}.

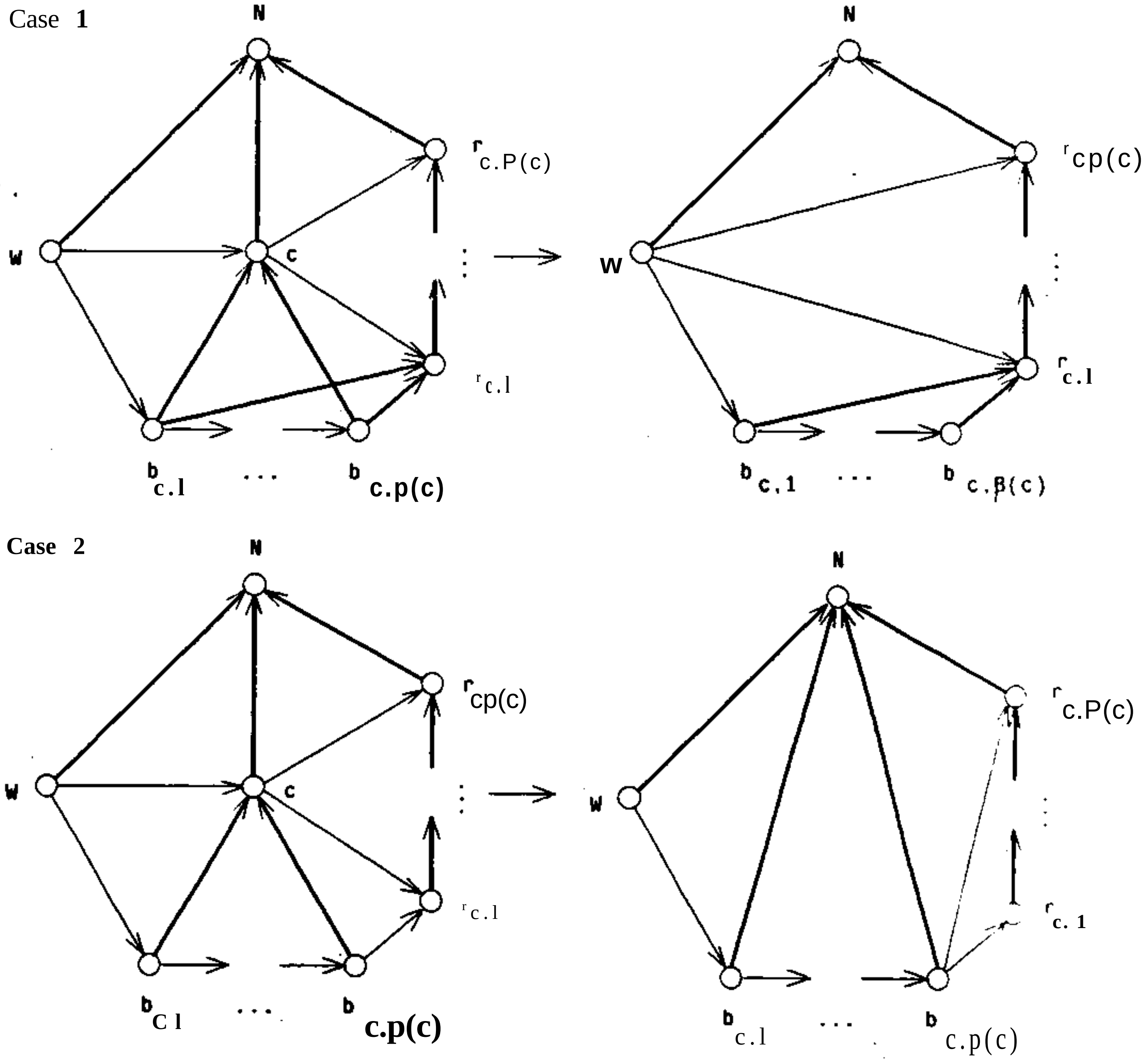

Figure 8: Lemma 7 - Transition from G_n+1 to G_n

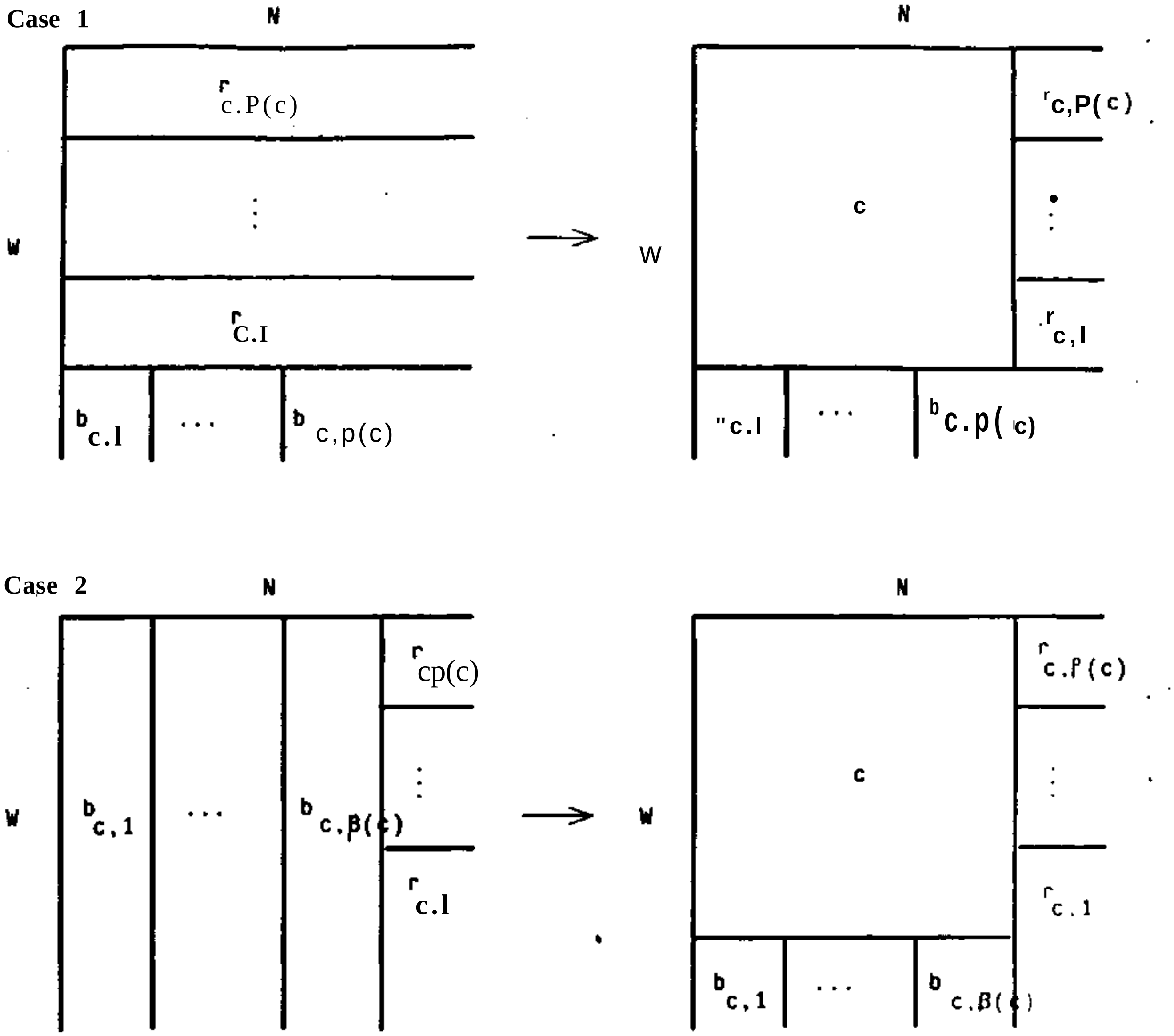

Figure 9: Lemma 7 - Transition from D_n to D_{n+1}

3 Generation of Orthogonal Structures

Let L_n be a loosely-packed arrangement of rectangles. A hole in L_n is *trivial* iff it can be eliminated by extending selected sides of certain rectangles in L_n parallel to the coordinate axes (without creating overlaps between pairs of rectangles). If every hole in L_n is trivial, L_n can be generated from a dissection D_n by reversing this process and by reducing the sides of certain components. Thus, if every hole in a loosely-packed arrangement of rectangles were trivial, the procedures developed for the generation of rectangular dissections could be used without major modifications for the generation of loosely-packed arrangements of rectangles.

This is, however, not the case. For example, the loosely-packed arrangement of rectangles, L_4, shown in Figure 4 contains a non-trivial hole, and the orthogonal structure representing L_4 consequently is not dense. It is, furthermore, easy to see that the only non-trivial holes that can occur in a loosely-packed arrangement of rectangles (after all trivial holes have been eliminated) are surrounded by four rectangles forming a pinwheel turning clockwise (as shown in Figure 4) or counterclockwise. The cases in which the premises of Lemma 6 are not met are standardized in the following and used to represent the spatial relations between rectangles forming precisely these types of non-trivial holes.

Note, at the outset, that if for a vertex, v, $a_{v,1}\mathbf{R}l_{v,\lambda(v)}$ but also $a_{v,1}\mathbf{a}l_{v,1}$, then $a_{v,1}\mathbf{A}l_{v,k}$ for some unique $k(<\lambda(v))$ and $a_{v,1}\mathbf{R}l_{v,k'}$ for $k'=k+1,\ldots,\lambda(v)$; similarly, if $l_{v,\lambda(v)}\mathbf{B}a_{v,1}$ but also $l_{v,\lambda(v)}\mathbf{r}\ a_{v,\alpha(v)}$, then $l_{v,\lambda(v)}\mathbf{R}a_{v,h}$ for some unique $h(>1)$ and $l_{v,\lambda(v)}\mathbf{B}a_{v,h'}$ for $h'=1,\ldots,h-1$. Analoguous results can be obtained if the premises of cases (ii) to (iv) in Lemma 6 are not satisfied.

An orthogonal structure is *restricted* iff it satisfies the following conditions for all vertices v.

$$\begin{array}{l}
\text{If } a_{v,1}\mathbf{R}l_{v,\lambda(v)} \text{ and } a_{v,1}\ \mathbf{A}l_{v,k} \text{ for some } k(<\lambda(v)) \\
[l_{v,\lambda(v)}\mathbf{B}a_{v,1} \text{ and } l_{v,\lambda(v)}\mathbf{L}a_{v,h} \text{ for some } h(>1)] \\
\text{then } w\mathbf{L}a_{v,1} \wedge u\mathbf{R}l_{v,k} \Rightarrow w\mathbf{L}u \text{ and } w\mathbf{A}v \wedge u\mathbf{B}l_{v,k+1} \Rightarrow w\mathbf{A}u \\
[w\mathbf{A}l_{v,\lambda(v)} \wedge u\mathbf{B}a_{v,h} \Rightarrow w\mathbf{A}u \text{ and } w\mathbf{L}v \wedge u\mathbf{R}a_{v,h-1} \Rightarrow w\mathbf{L}u].
\end{array} \tag{11}$$

Under condition (11), the spatial relations among the rectangle corresponding to vertex v, $f(v)$, and other rectangles surrounding a non-trivial hole at the upper left corner of $f(v)$ are defined according to the construction used in the proof of Theorem 1 (see Figure 10). The first part of (11) describes rectangles forming a pinwheel that turns counterclockwise, while the second part describes rectangles forming a pinwheel that turns clockwise. Based on this observation, the following theorem is easy to prove.

> **Theorem 8:** Every loosely-packed arrangement of rectangles, L_n, is represented by a restricted orthogonal structure, G_n.

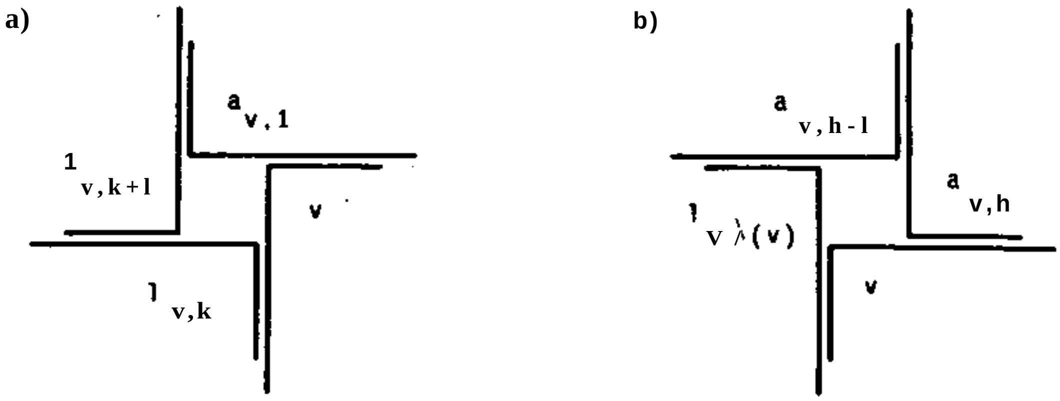

Figure 10: Condition (11) - Illustration

To select vertex v as a reference in the formulation of condition (11) was an arbitrary choice. The following lemma shows that each of the other three vertices representing a rectangle involved in the formation of a non-trivial hole could have been used.

Lemma 9: Each of the the following conditions is equivalent to (11).

> If rv,P(v)B^v,a<v) an<* rvpi v)Rav. h for some h(< a(v))
> k a (v) k p (v) and av.a(v)Arv, / for some /(< p(v))J,
> then w\rv p(v)AuBayji = > wAw and wRvA wLav ^+1 = >H>RW
> [wRav ,a^Y)AuLrv i => vvLwand wAvA uBrvj+i => w\u].
>
> If 6V, 0(v)Lrv, j and 6V, £(v)Brv, ; for some /(> 1)
> K l A6v./3(v) an<* rVt i R6v,y for somey(<fi(v))],
> then H'Rftv^v) A wLrv, 7 => n/Rwand vvBvA wArv ,_i => wBu
> [>vBrvj Au\bvj => wBwand >vRvA wL6Yy+1 => wRu].
>
> If /Vf 1 \bv ,i and /v, i Lbvj for some y(> 1)
> [bYt i R/Vt i and 6V, i B/v, ^ for some k(> 1)],
> then H>B/V i Au\bvj => wBwand wLvA uRbVfy_i => >vLw
> [wLbv,1 ∧ uRlv,k => wLu and wBv ∧ uAlv,k_i => wBu].

Proof: (11) => (14). l.ciGn be an orthogonal structure so that (11) holds for every vertex ot < >. Let furthermore v' be a vertex of Gn so that // i A/y. \ and // \ l i / ^ for some /(> 1). I VP. •;. bvf i by v. Suppose /vX(v)B</v ,. If /r.\u)Btfv<a<v). Lhcn y/^/v.A(v) (h>' Lemma 6) and connected to 6/,/- i by a vertical path longer than one. which is impossible. If !v \(v)\< some then av i,= v' or fl^/Jr7. Uv (11). H'LVAWR(/V/,.] = >K1W. Consequently, bv'j_\\v'. whk again impossible. ITius. /,.A, v)B(/V j. and by Lemma 5. tfVtiR/v.X(v> furthermore, v^i/, K. (11) then.

> wLv' ∧ uRbv',j-1 => wLu and uAbv',1 ∧ wBlv',1 => wBu.

(14) =>(13).(13) = >(12)and(12) =>(11)can be provedanaioguously.

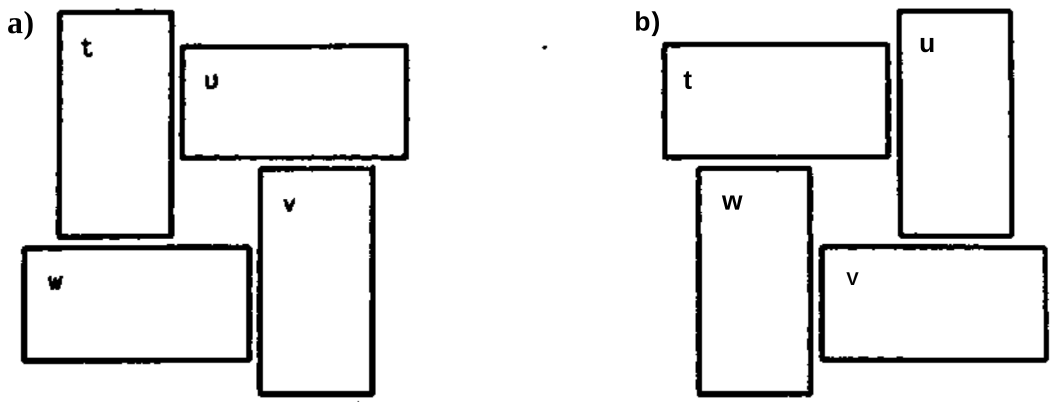

Figure 11: Rectangles forming a pinwheel turning a) clockwise and b) counterclockwise

Let L_n be **a loosely-packed** arrangement of rectangles and G_n a restricted orthogonal structure represenung G_n. **Suppose** /,*u,* v**and** *ware* vertices in G_n representing four **rectangles** in L_n that form **a** pinwheel around a non-trivial hole. If the pinwheel turns counterclockwise (see Figure 11 a).

$$t = l_{u,1},\ u = a_{v,1},\ v = r_{w,\beta(w)} \text{ and } w = b_{t,\beta(t)}.$$

Furthermore,

$$a_{v,\alpha(v)} = a_{w,\alpha(w)},\ b_{t,1} = b_{u,1},\ l_{u,\lambda(u)} = \text{KMv)} \text{ and } r_{w,1} = r_{u,1}.$$

If **the** pinwheel turns clockwise (see Figure 11 b),

$$t = a_{w,\alpha(w)},\ u = r_{t,1},\ v = b_{u,1} \text{ and } w = l_{v,\lambda(v)}.$$

and

$$a_{w,1} = a_{v,1},\ b_{u,\beta(u)} = b_{t,\beta(t)},\ l_{v,1} = l_{u,1} \text{ and } r_{t,\rho(t)} = r_{\text{lv},\rho(\text{»})}.$$

These observations show that (11) standardizes the spatial relations holding among rectangles in the . . of **a** non-trivial hole in a natural and - more importantly - predictable way. ÍTiis proves extreme! •. ^ during the development of a generator for restricted orthogonal structures.

For the present context, two types of generators arc of interest: (a) those that enumerate all non-i^omor*(Drs*ic restricted orthogonal structures G_n for a given //: and (b) those that also distinguish for a given G_n dif \ created by different assignments of labels to the internal vertices of G_n. The first type of procedures is interesting for theoretical purposes, while the second type is important mainly for practical jppiicatinns where labels arc used to distinguish different functional units. The present paper concentrate o* the generation of non-isomorphic orthogonal structures and presents a grammar that can be used to this cr :

Figure 12 shows the rules of the grammar. Hach mlc is a recursive re-write rule consisting of a left-hand and a right-hand side both of which are parameterized. A rule can be applied to a restricted orthogonal structure G_n iff there exists an assignment of values to the parameters of its left-hand side under which this side becomes a subgraph of G_n. The application itself produces a directed graph G_n+ j by substituting the right-hand side of the rule (under the same assignment of values) for its left-hand side in G_n. The grammar is complete by adding as an initial configuration the orthogonal structure (7j which is dense (and therefore restricted).

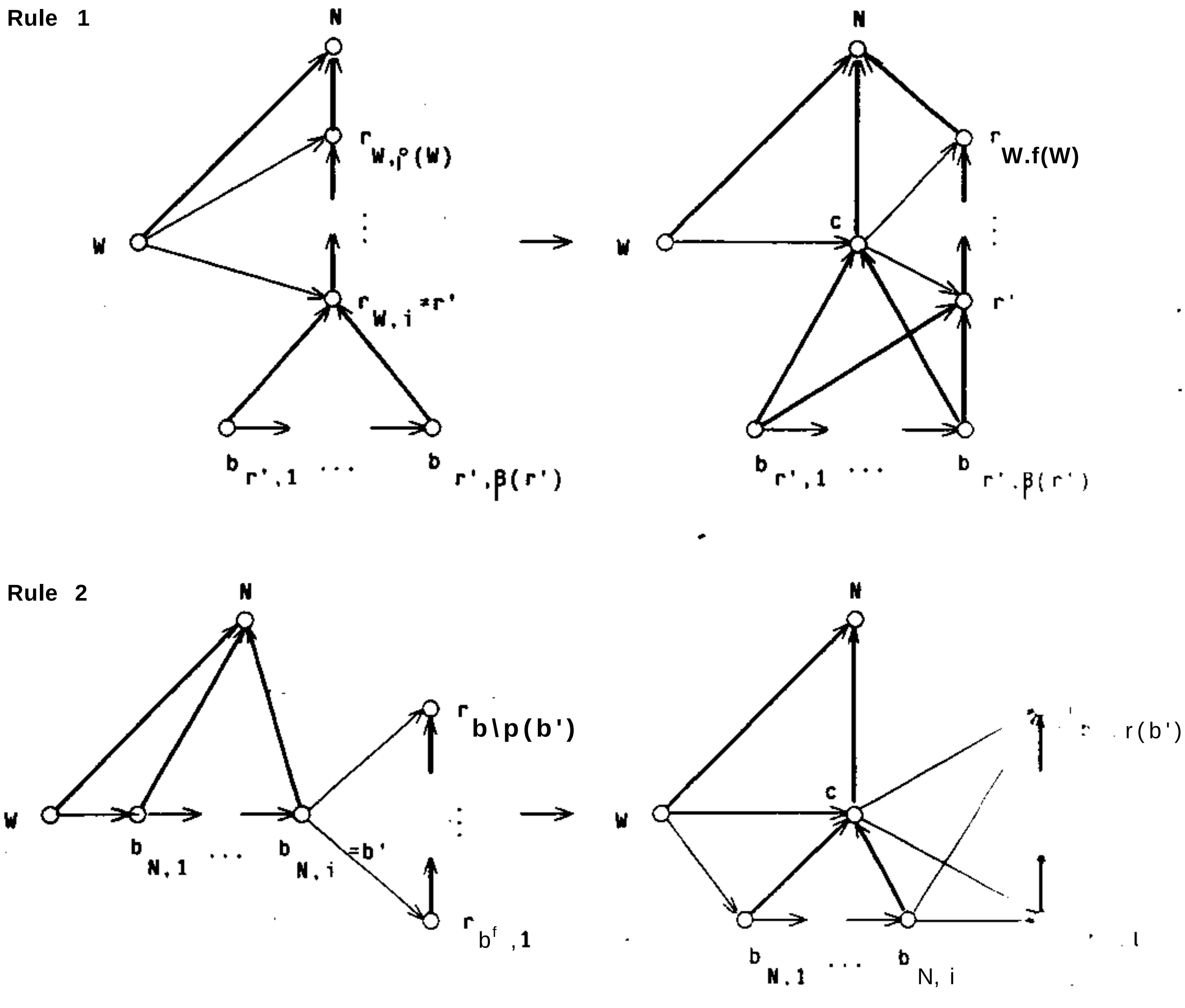

Figure 12: Rules 1 and 2

All rules follow the basic approach that has been used before in a similar context: vertices arc insert

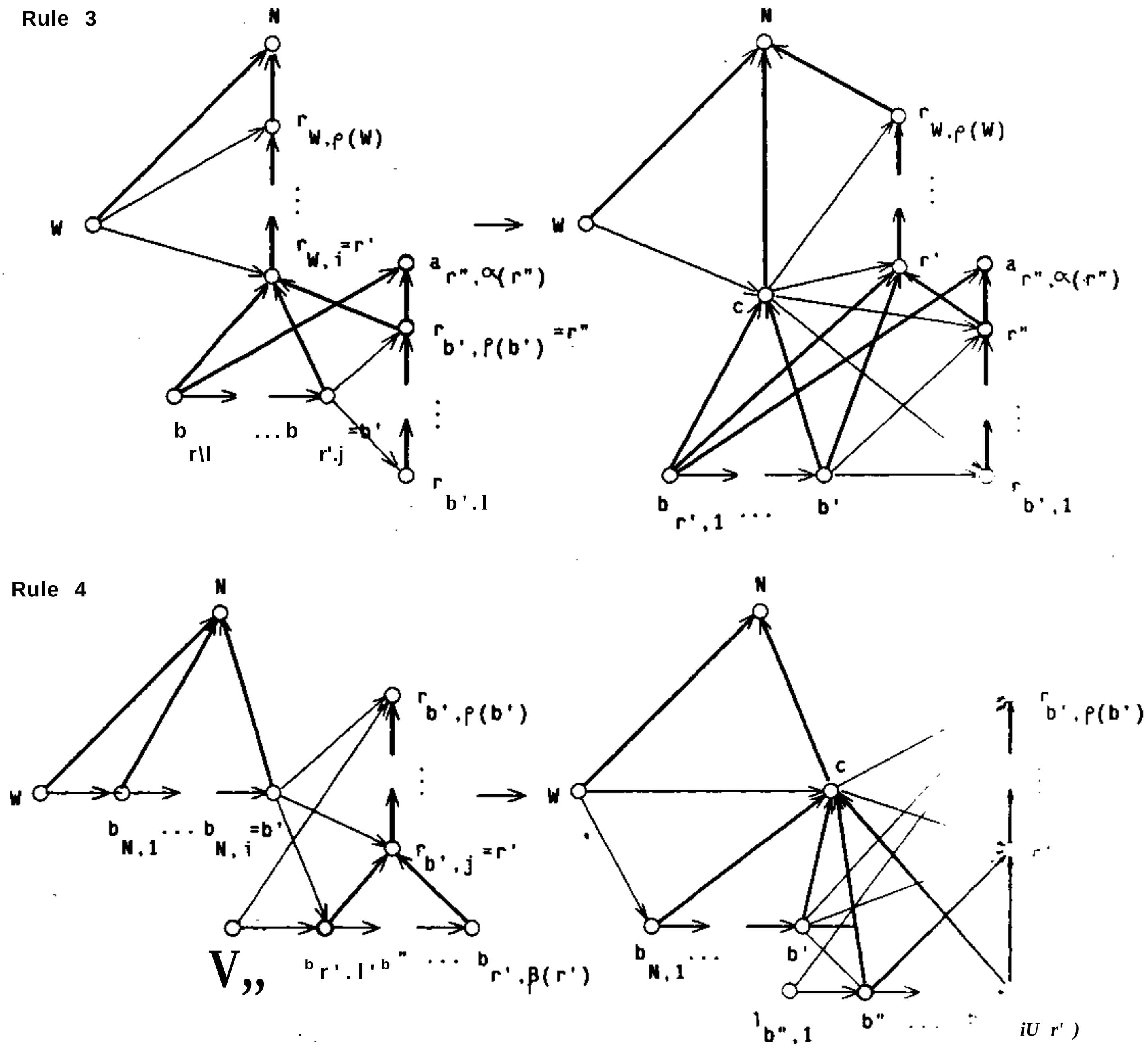

Figure 12 (continued) - Rules 3 and 4

upper left corner*, which makes it particularly easy to avoid duplication of isomorphic struct.. Corollary 12). Rules 1 and 2 generate only dense structures, while rules 3 and 4 generate ^:: representing arrangements with non-trivial holes. TTic hole always occurs at the lower right conk" rectangle corresponding to vertex (; it turns counterclockwise in case of rule 3 and clockwise in uise « · A geometric interpretation of the application of rules 1 to 4 is shown in Figure 13.

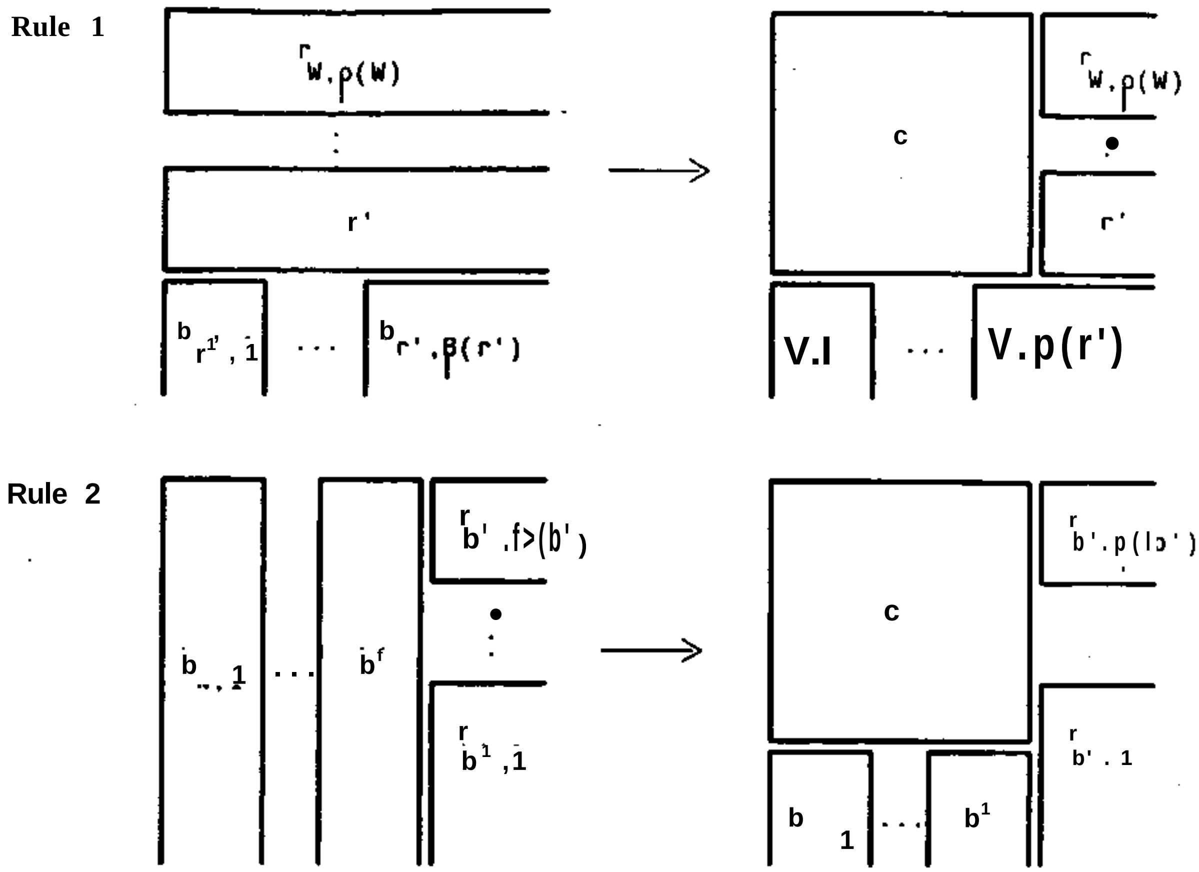

Figure 13: Rules 1 and 2 - geometric interpretation

Theorem 10: (Closure) If G_n is generated from G_f through **a** series of applications of rules 1 *u* -
G_n is **a** restricted orthogonal **structure.**

Proof: Only rules 1 and 2 **are applicable to** *G* they produce **the** two non-isomorphic d
orthogonal structures shown in Figure 14.

Suppose G_n is **a** restricted orthogonal structure to which rule 3 is applicable, generating a stria: G_n+\. Observe that *W* is the only vertex directly to the left of vertex *c* and *N* is the onh w directly above *c*. For c\ery vertex z so that *zbc*. *zbb'* or *zW*. For every vertex *z'* so that r z'aZ/ or *z'xb'*. Thus. (6) is satisfied for c and any other vertex of G_n+ \. The condition also h for all other pairs of vertices.

Suppose (/„+ j contains an v-colorcd arrow pointing form *a* to *b* and a directed, .v-colorcd p^i' from *a* to *b* longer than one. Since *(i$_n$* is an orthogonal structure. *A* or *P* cannot be in Consequently, cither *a* or *b* is the vertex *c*. But the construction of G_n+\ from the orthoi;- stniclurc G_n assures that if < is connected to another \ertex by an arrow, it is not connected u> \crtcx by a directed, uniformly-colored path longer than one. Thus. *(i$_n$*+\ satisfies (' obviously satisfies (S) and (9). *(i$_n$*+ j therefore is <m orthogonal structure.

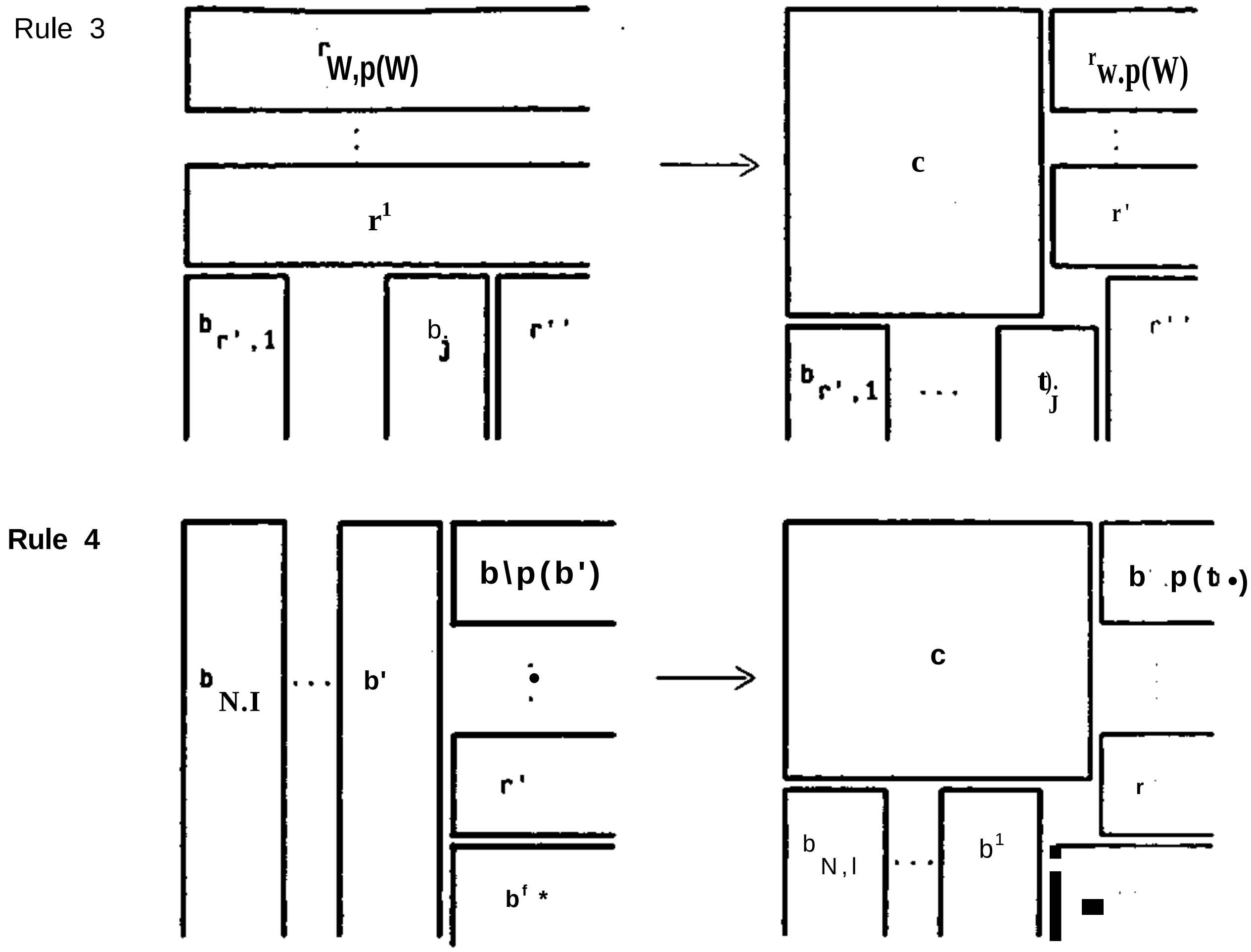

Figure 13 (continued): Rules 3 and 4 - geometric interpretation

. Observe that if u is a vertex in (7„+! not incident with c\ the vertices incident with u rcm.i unchanged in the transition from G_n to $G_n+_{\{}$. and u satisfies (-11). Any vertex v s<^ :,h v€{c, $b/_tf$, $r_wj\j'<j\i'>i\}$ cannot satisfy the premise of (11).

Consider vertex r''. In $G_n+\%$, a_r» $i(\sim r')Rl_r''$^/'{^c) and $a/'_ti\l_r''_{tk}(=b')$. vvh, *=\(r//)—1. Thus, r'' satisfies the premise of (11). By construction, $uRl_r''_k$ = >cLu, and MFK,- is the only venex directly to the left $ofa/'_t\%$

$$wLa_r''j \wedge uKl/f_k \Rightarrow wLu.$$

Furthermore, $b/1$ B<;//$_{a(r//)}$ and consequently, $w\wedge r''$ = >$w\b/_tji$ for / = 1____/. !: construction, $u\\a_r$» \ = >$u\\c\{$ ≡/$_r$'' $_A$.+ j). Thus,

$$w\r'' \wedge u\ RI_r''_{tk}+1 \Rightarrow w \wedge w,$$

and r" satisfies (11).

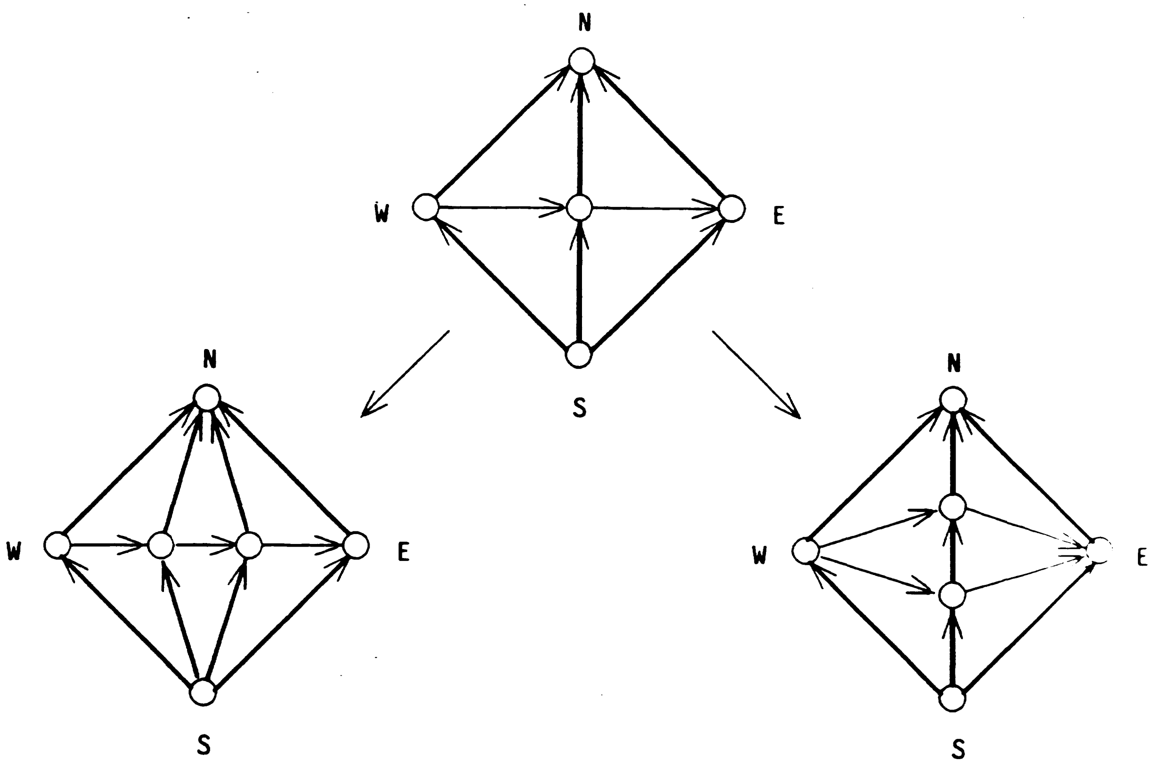

Figure 14: Generation of two structures S_2 from S_1

For every vertex w so that $w \in \{r_{b',i'} \mid i' < \rho(b')\}$, $a_{w,1} \mathbf{R} l_{w,\lambda(w)}$ in both G_n and G_{n+1}, and the incidences between $a_{w,1}$ and the other vertices directly to the left of w in G_n remain unchanged in the transition from G_n to G_{n+1}. Since w satisfies (11) in G_n, it also satisfies (11) in G_{n+1} and G_{n+1} is restricted.

•

In a similar way, it can be shown that an application of rules 1, 2 or 4 to G_n generates a restricted orthogonal structure G_{n+1}. The theorem then follows by induction on n.

Theorem 11: (Completeness) Every restricted orthogonal structure G_n is generated from G_1 by a series of applications of rules 1 to 4.

Proof: (Sketch). Let G_n be a restricted orthogonal structure. Observe that the last vertex directly to the right of W, c, is the second vertex directly below N. The configuration of arrows incident with c must belong to exactly one of the cases depicted as the right-hand sides of rules 1 to 4 in Figure 12. By applying the appropriate rule 'backwards', a restricted orthogonal structure G_{n-1} is generated. This fact suggests an inductive proof of the theorem.

Corollary 12: The sequence of rule applications generating an orthogonal structure is uniquely determined.

At the present time, a generator based on rules 1 to 4 is under development at the Computer-Aided

Laboratory of the l^cpartmcnt of Architecture at Carncgic-Mcllon University. A pilot version, called LOOS, has been implemented and can be used to generate non-isomorphic restricted orthogonal structures. Rather than showing these structures directly to the user (who might have a hard time trying to understand them), the program derives, for each structure it has generated, a loosely-packed arrangement of rectangles represented by that structure and displays this arrangement to the user, **llic** arrangements selected contain only non-trivial holes, thus enabling the user to deduce the underlying structure without ambiguity (by simulating the process employed in the proof of Theorem 1).

Examples of such **arrangements are** shown in Figure 15. The first of **these** is **represented** by a structure to which **all rules can be applied, and rules 1** to **3 can be applied under more than one assignment of** parameters. The results of **these applications are illustrated by the remaining arrangements shown in Figure 15.**

LOOS was also used to count the number of non-isomorphic restricted structures with up to 10 internal vertices; the results are listed in Table 2.

n	number of non-isomorphic, restricted structures G_n
1	1
2	2
3	6
4	24
5	116
6	642
7	3,938
8	26,194
9	186,042
10	1,395,008

Table 2: Enumeration of non-isomorphic restricted structures

The grammar implemented through LOOS can be extended in a straight-forward way to also generate ail assignments of labels to the interior vertices of an orthogonal structure. This can be done cither by generating all permutations of assignments after a complete structure has been produced: or by assigning the labels thai have not been assigned yet to the vertex c inserted during each application of a rule. Roth methods hj disadvantages. The first method makes it impossible to use constraints (which arc usually specific to the objects that arc allocated) for pruning the search tree during the generation, a feature that is indispcnsihk r. r efficient searches. In the second method, the order in which the objects arc allocated changes between LCTL branches, which makes it difficult for users to follow the process (it also creates inefficiencies because in

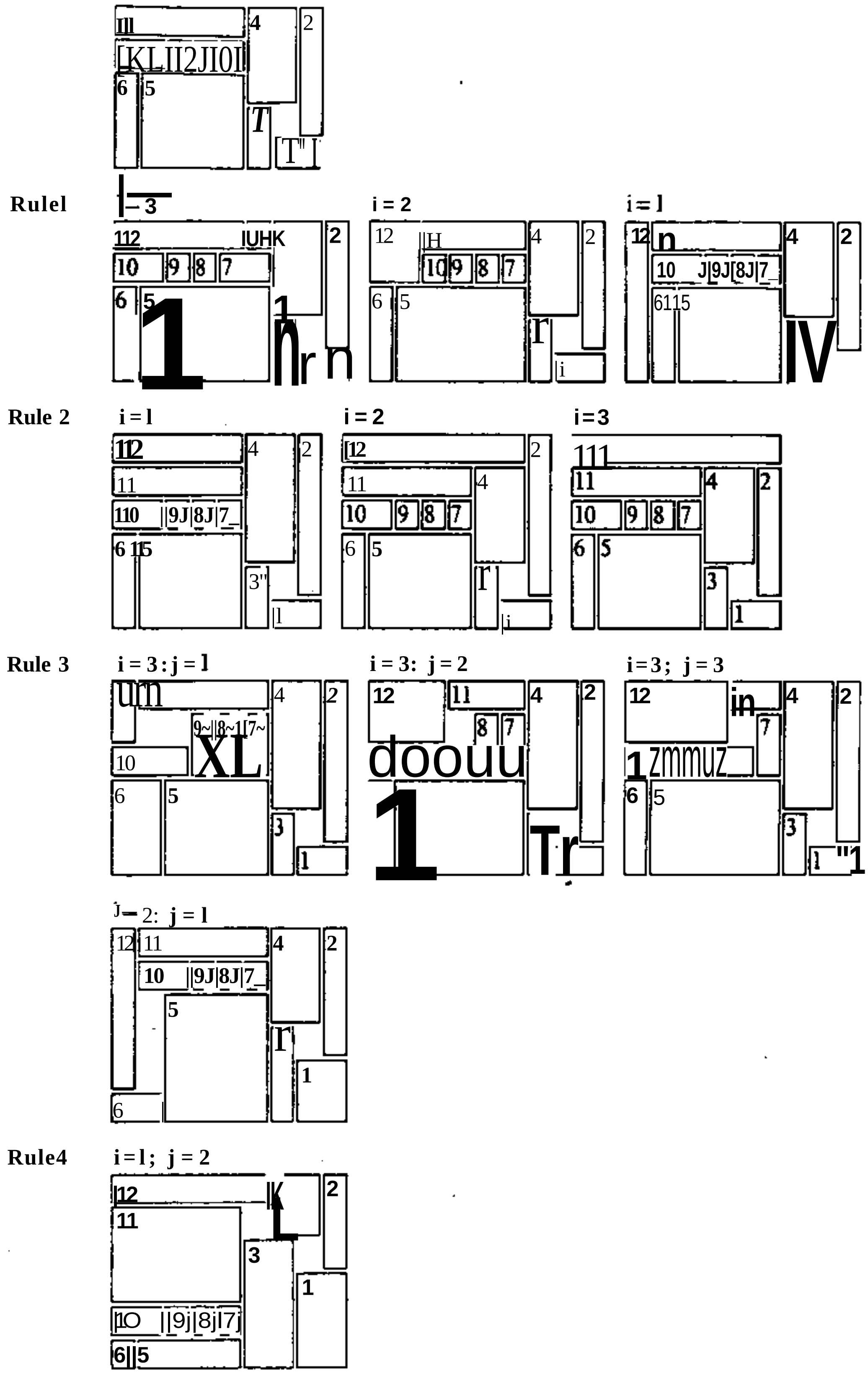

Figure 15: Program LOOS - sample output

higher constrained objects should be allocated first). In order to allocate labelled objects in a given (user-defined) order, the grammar must be expanded to allow for an allocation not only in the upper left comer, but everywhere in the developing configuration.

The design of such a grammar has a precedent in the grammar described in [5] for the generation of rectangular dissections with labelled components. Kach comer of such a component can be formed by two walls in exactly two ways; thus, there exist $2^4 = 16$ possibilities for forming the four corners of a component. In order to generate these configurations, 16 rules are needed at the outset ITie quoted paper shows, however, that the number of required rules can be reduced under suitable parameterizations. Given this precedent, the task to develop an expanded grammar for the generation of restricted orthogonal structures with labelled vertices, where the vertices enter the process in a fixed order, is clear-cut There are four possibilities for the configuration of rectangles at a corner of a given rectangle in a loosely-packed arrangement of rectangles. This results in $4^4 = 256$ possibilities for the configuration of rectangles at the four corners of that rectangle and in the same number of possibilities for the vertices incident with the corresponding vertex in a restricted orthogonal structure. Thus, 256 rules are needed at first to generate these possibilities. To make the task of designing and testing these rules more manageable, parameterizations are desired that drastically reduce the number of required rules. The specification of these rules will be the topic of a second paper.

4 A Generative Expert System for the Design of Architectural Layouts

So far, the discussion has focusscd on the syntactic properties of loosely-packed arrangements of rectangles and their representations. But for each problem domain, the rectangles being allocated have a specific meaning, and specific constraints, criteria, guidelines or rules govern the allocation of these objects. To avoid confusion, ^two^ types of rules should be distinguished in this context: *generative* rules, such as the ones described in the previous section, which create representations of layouts; and *diagnostic* rules used to evaluate a layout Among the latter, it is useful to further distinguish rules that evaluate *constraints* (i.e. properties a solution must posess if it is to be considered acceptable or feasible) from those that evaluate *criteria* (i. e. properties that make certain layouts better than others).

The incorporation ofefiagnostic rules into the generation process isjndispensiblejbr several reasons. These rules assure, first of all, that the generated objects represent meaningful solutions to the problem at hand; in addition, Ihey can be used to filter out the less promising solutions so that users are presented Aiih a manageable set of alternatives for further analysis and evaluation. The rules might also provide an important device to prune the search tree so that the computations involved become feasible.

Section 1 argued that many diagnostic rules are never stated explicitly in a problem description, nor h^ve *[hey* been systematically documented and collected anywhere. They are part of the general expertise of designer which is acquired over years of practice and remains unarticulated unless designers are forced to cxpl.un the faults of a particular solution. The section also introduced expert systems as vehicles to extract exact!/ this type of knowledge from experts and to encode it explicitly in the form of a knowledge base, which i^used by the system to solve problems within their field of expertise. This knowledge base is built up through-& sequence of iterations in which experts observe the performance of the program; criticize its performance inspect the knowledge that has been used; and suggest additions or modifications to the knowledge base facilitate this process, expert systems are programmed in a style that contrasts with the traditional, "pn ... approach by making changes in the knowledge base as easy as possible.

Section 1 suggested that the performance of layout generators can improve if they take on the to... expert system. A precedent that appears particularly interesting within this context is the DENDRAL program, an expert system that predicts the chemical structure of certain compounds from their m.i^ and specifications (by type and number) of the atoms in the compound (see [2] or [1] for dcscripiu"^ program). The core of the program consists *o(* a *gcnzralsir* that produces candidate structures * hk h tested against the given spectrum; the structures most likely to have a spectrum of that kind arc p.isv : user (in a ranked order). The structures arc represented as planar graphs; and the closure and u»mpk«"U*v* of the process (with respect to the set of possible chemical structures) has been proved mathematics.^.

space searched by the generator is restricted by *ajlgnnez* a program which infers constraints from the input data in order to reduce the (possibly large) number of structures that must be generated (and subsequently tested), llie *tester*, in turn, accepts a candidate structure from the generator, predicts its mass spectrum and compares it to the given spectrum. Unlike the generator, the tester is not based on a complete and systematic theory, but reflects the various bits of knowledge expert chemical analysts bring to bear on this task. ITie program was used over a long scries of test runs to discover and encode this knowledge in the manner described above (a detailed account of this process is given in [3]).

The remainder of this section will outline an expert system for the design of architectural layouts which is loosely modelled after DENDRAL. The major components of the system arejLgenerator and ajesLci_&hich are again developed by contrasting modes of reasoning: the generator is completely specified by a deductive theory established *a priori,* while the tester is to be built up inductively over a series of applications.

The generative rules described in the previous section (or an expanded version of these rules) can form the basis of a generator able to produce all possible orthogonal structures with a given number of \en ices to represent loosely-packed arrangements of the same number of rectangles. The rules by themselves, however. do not completely specify this generator. Any intermediate structure generated by a series of rule applications can normally be expanded by more than one rule and under various assignments of values to the parameters in the rule. In order to select a particular application, a *control strategy* is needed which, together with the initial configuration, completes the specification of the generator.

Such a control strategy is easy to define if the solution set is to be completely enumerated. In this case c en, intermediate structure must be expanded in every possible way, a process that can be implemented n * straight-forward manner as depth-first search through an appropriately constructed search tree whose nodes correspond to the intermediate or terminal solutions that are generated. The program DIS, for exampie, uses this type of control strategy. The same program has also shown that the search tree must he p extensively if problems calling for the allocation of even a moderate number of objects are to become te s- The generative rules used by DIS, as well as the rules shown in the previous section, make it possible certain problem constraints or criteria for this purpose. For it can be shown that many constraints and . that are not satisfied by an intermediate solution cannot be satisfied by any solution derived from u intermediate solutions therefore do not have to be expanded. ITic program DIS uses only r^ adjacencies to this end. ^strategy which becomes efficient if enough of these constraints are _• Constraints that can only be evaluated if the layout is complete arc formulated for each terminal *node* used to further reduce the number of alternatives (examples of this type of constraints are m•» dimensions or areas which can onl> be satisfied in certain densely-packed arrangements if a sufficient n of objects has been allocated). Still, the number of alternatives presented to the user can remain Linic morcjJiorough^selection is needed, taking into account not only constraints, but also critcjJaJtoiiUiclp to find the better solutions among the feasible ones.

I suggest that at least for a pilot system, this purpose can be achieved by a branch-and-bound strategy which expands those and only those intermediate solutions which are at least as good as any other solution generated before (independent of its depth in the search tree). To evaluate a node, I suggest using a simple evaluation function which computes for each node, s, the triple $<p(s) = <c(s),d(s),e(s)>$, where $c(s)$ is the number of constraints, $d(s)$ the number of'strong' criteria and $e(s)$ the number of'weak' criteria violated by s. Based on these triples, solutions can be ranked lexicographically'.

An example of a diagnostic rule evaluating a constraint is the following:

> The sum of the minimum horizontal dimensions of any sequence of objects arranged from left to right between W and E cannot exceed the maximum horizontal dimension of the available area.

Clearly, this constraint can be used to prune the search tree provided that the addition of a new object docs not alter the spatial relations between the objects that are already allocated (the rules presented in the previous section guarantee this property.) Experiments with DIS have shown that this type of constraint is one of the most frequently violated dependent constraints; it should therefore provide for an effective pruning device. (Incidentally, this constraint also demonstrates how dimensional considerations can enter the generation process at early stages.)

If the given design problem deals with the remodelling of a kitchen, the following rule might evaluate ,\ vtnmg criterion:

> The sink should be adjacent to the existing plumbing stubs.

For the same type of problem, the following rule might be used to evaluate a weak criterion:

> If the main food preparer is right-handed, the range should be to the right of the main VMU surface (when viewed from the front).

The challenge is to design and implement a tester which returns the proper value of the objccti\e t•,re- taking a broad range of constraints and criteria into account which, at the outset, arc not known, but added to the evolving knowledge base with case, preferably through declarative statements of the tvpc *• above.

Acknowledgements

The opportunity to work out the ideas expressed in the present paper was given to me during a rccem the Centre for Configurational Studies. The Open University. Milton Kcyncs. Knghind. I am pin indcptcd to Ph. Stcadman. the Centre's director, for his invitation and support and to C. Harl for numerous discussions and suggestions. I would also like to acknowledge contributions by J. Carrcra and T. Gla\in, who implemented the layout generator LOOS at the Computer-Aided Design Laboratory, Department of Architecture, Carnegie-Mellon University.

References

[1] Buchanan, Bruce G. and Feigenbaum, Edward A.
Dendral and Meta-Dendral: their applications dimension.
Artificial Intelligence 11:5-24, 1978.

[2] Buchanan, B.; Sutherland, Georgia and Feigenbaum, E.A.
HEURISTIC DENDRAL: a program for generating explanatory hypotheses in organic chemistry.
In Meltzer, B. and Michie, D. (editor), *Machine Intelligence 4*, pages 209-254. Edinburgh University Press, Edinburgh, 1969.

[3] Buchanan, B. G.; Sutherland, G. L. and Feigenbaum, E. A.
Rediscovering some problems of artificial intelligence in the context of organic chemistry.
In Meltzer, B. and Michie, D. (editor), *Machine Intelligence 5*, pages 253-280. Edinburgh University Press, Edinburgh, 1970.

[4] Earl, C. F.
Rectilinear shapes.
Environment and Planning B 7:311-342, 1980.

[5] Flemming, U.
Wall representations of rectangular dissections and their use in automated space allocation.
Environment and Planning B 5:215-232, 1978.

[6] Flemming, U.
Wall representations of rectangular dissections: additional results.
Environment and Planning B 7:247-251, 1980.

[7] Galle, Per.
An algorithm for exhaustive generation of building floor plans.
Communications of the ACM 24:813-825, 1981.

[8] Gero, John S. and Coyne, Richard.
The place of expert systems in architecture.
In *PARC 83. Proceedings of the International Conference on Computers in Architecture*, pages [illegible]. Online Publications Ltd., Pinner, Middlesex, UK, 1983.

[9] Grason, John.
A dual linear graph representation for space-filling location problems of the floor plan type.
In Moore, Gary T. (editor), *Emerging Methods in Environmental Design and Planning*, pages [illegible]. M.I.T. Press, Cambridge, Mass., 1970.

[10] Mitchell, W. J.; Steadman, J. P. and Liggett, Robin S.
Synthesis and optimization of small rectangular floor plans.
Environment and Planning B 3:37-70, 1976.

[11] Radford, Antony D. and Gero, John S.
Tradeoff diagrams for the integrated design of the physical environment in buildings.
Building and Environment 15:3-15, 1980.

[12] Rychcner, Michael D.
Hxpcrt systems for engineering design: problem components, techniques and prototypes.
Report DRC-05-02-83, Design Research Center, Carncgic-Mcllon University.
1983.

[13] Steadman, Philip.
The automatic generation of minimum-standard house plans.
Paper delivered at the Second Annual Conference of the Environmental Design Research Association.
1970.