Environment and Planning B, 1979, volume 6, pages 331-352

3-rectangulations: an algorithm to generate box packings

R Krishnamurti
Centre for Configurational Studies, The Open University, Milton Keynes MK7 6AA, England
Received 23 August 1979

Abstract. 3-rectangulations are spatial representatives of assemblies of boxes into a box. Algorithms to generate various classes of 3-rectangulations are developed. The method is extended to specify the generation of higher-dimensional d-rectangulations, $d \geqslant 4$.

3-rectangulations

I consider the following combinatorial problem: how many arrangements of boxes into a box are there subject to the proviso that they be 'spatially distinct'? Within an architectural context an assembly of boxes into a box may be regarded as 'a three-dimensional building description', and may be seen as the natural extension of the 'floor-plan problem' which has been discussed elsewhere (see, for instance, March and Earl, 1977; Earl and March, 1979; Flemming, 1978; Krishnamurti and Roe, 1978). The solution presented here takes the form of a generative scheme or an algorithm which enumerates equivalent families of box packings defined in terms of line designs on a three-dimensional grid, namely the 3-rectangulations. The definitions and notations that appear herein are consistent with those for 2-rectangulations (Earl, 1978; Krishnamurti and Roe, 1978). The material presented in this paper is sufficiently self-contained.

A *3-rectangulation* is an arrangement of an (l, m, n) unit grid into nonoverlapping cuboids whose faces lie on grid planes. The faces are combined to form *maximal planes* (Earl, 1978). (A set of maximal planes is a set of plane segments the union of any two of which never forms a single plane segment.) Maximal planes may be defined by a set of boundary lines. Constructively a 3-rectangulation may be viewed as a finite set of maximal planes each of which is coincident with one of the grid planes. Of these, six share boundary lines (those of the bounding cuboid) whereas the rest have boundary lines coincident with other maximal planes but do not share boundary lines. Figure 1 illustrates this construction.

3-rectangulations may be classified into the following type categories.

1. A 3-rectangulation is *proper* if $l, m, n \geqslant 2$.
[An improper 3-rectangulation may be considered as a solid 2-rectangulation of unit height.]

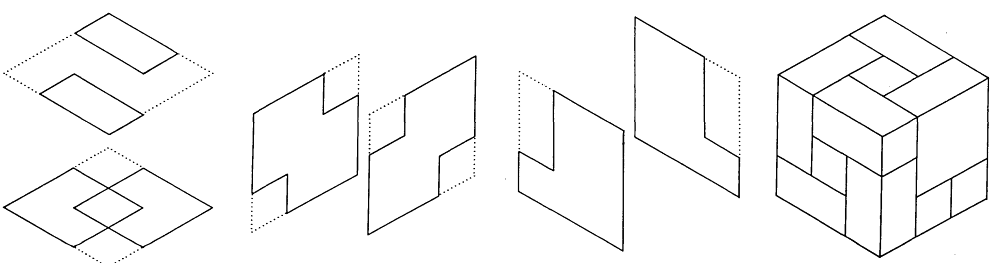


Figure 1. The internal maximal planes of a 3-rectangulation.

2. A 3-rectangulation is *trivalent* if each cross section is a trivalent 2-rectangulation. [A cross section of a 3-rectangulation is a plane section parallel to one of the grid planes and which does not contain any maximal planes. Clearly each of the $l+m+n$ possible cross sections corresponds to a packing of rectangles into a rectangle, or a 2-rectangulation.]

3. A 3-rectangulation is *nonaligned* if each grid plane contains at most one maximal plane.

4. A 3-rectangulation is *standard* if each grid plane contains at least one maximal plane.
[For standard 3-rectangulations it is clear that every grid plane must contain a face of one of the constituent cuboids.]

5. A 3-rectangulation is *fundamental* if it is standard, trivalent, and nonaligned. [The name arises on account of its connection with 'fundamental architectural arrangements' described in March and Earl (1977).]

The number of component cuboids of a 3-rectangulation is its *content*. A 3-rectangulation with content p is explicitly referred to as a $[p, 3]$-rectangulation. Two 3-rectangulations are *equivalent* if they are identical after a sequence of 'right-handed' rotations or reflections. Enantiomorphs, or right-handed and left-handed versions of the same object, are regarded as distinct arrangements. The proviso 'spatially distinct' corresponds to the concept 'standard and nonequivalent'. In this paper the enumeration of nonequivalent standard proper $[p, 3]$-rectangulations, for p fixed, via the counting of the representatives—termed *canonical 3-rectangulations*—of the equivalence classes is considered.

Canonical 3-rectangulations

In this section a representation for 3-rectangulations is developed. This representation enables us to define a condition that uniquely determines a representative for each equivalence class. First we need to consider the (l, m, n) unit grids upon which the representation is based.

6. Earl (1978) has shown that the dimensions of (l, m, n) unit grids for which there exists at least one standard $[p, 3]$-rectangulation must satisfy

$$l+m+n-2 \leqslant p \leqslant lmn \ . \tag{1}$$

From symmetry considerations (which are explained later in this section) we may stipulate that

$$l \geqslant m \geqslant n \geqslant 2 \ . \tag{2}$$

The following is the set of integer triples (l, m, n) that satisfy inequalities (1) and (2):

$$\Lambda = \left\{(l, m, n)\colon\ 2 \leqslant n \leqslant \left\lfloor \frac{p+2}{3} \right\rfloor,\ n \leqslant m \leqslant \left\lfloor \frac{p+2-n}{2} \right\rfloor,\right.$$
$$\left. \max\left\{m,\ \left\lceil \frac{p}{mn} \right\rceil\right\} \leqslant l \leqslant p+2-n-m \right\},$$

where, for any real number r, $\lfloor r \rfloor$ denotes the greatest integer less than or equal to r and $\lceil r \rceil$ denotes the least integer greater than or equal to r.

For each $(l, m, n) \in \Lambda$ we can define the following grid quantities.

7. The grid cells are associated with unique integer coordinates (x, y, z), where $x \in \mathsf{Z}_l$, $y \in \mathsf{Z}_m$, and $z \in \mathsf{Z}_n$. The internal grid planes lie on the planes $X = x \in \mathsf{Z}_l^+$,

$Y = y \in \mathsf{Z}_m^+$, and $Z = z \in \mathsf{Z}_n^+$. ($\mathsf{Z}_k = \{0, 1, ..., k-1\}$ and $\mathsf{Z}_k^+ = \mathsf{Z}_k - \{0\}$.) The grid plane $X = x_0$ separates the sets of grid cells $\{(x_0 - 1, y \in \mathsf{Z}_m, z \in \mathsf{Z}_n)\}$ and $\{(x_0, y \in \mathsf{Z}_m, z \in \mathsf{Z}_n)\}$. Any internal grid plane may be considered as a set of grid faces each of which may be assigned integer coordinates. As an example the grid face (x_0, y, z) on the plane $X = x_0$ separates the grid cells $(x_0 - 1, y, z)$ and (x_0, y, z), which in turn share the grid face.

8. The *threading pattern*, Γ, is the bijection between the grid cells and the integers $1, 2, ..., lmn$ denoted by the ordered set

$$\langle\langle (x_1, y_1, z_1), (x_2, y_2, z_2), ..., (x_{lmn}, y_{lmn}, z_{lmn}) \rangle\rangle ,$$

where

$$\Gamma(x_q, y_q, z_q) = lmz_q + ly_q + x_q + 1 = q .$$

The threading pattern is a particular numbering of the grid cells. In our case the cells have been labelled according to increasing x-, increasing y-, and increasing z-coordinates in that order. Theoretically any bijection between the grid cells and the integers $1, 2, ..., lmn$ may be chosen as the threading pattern. The one given here is convenient for describing the generative scheme for 3-rectangulations.

9. The *cell type* is the mapping of the grid cells onto the integers $0, 1, ..., 7$ defined by the following Boolean truth table:

cell type	0	1	2	3	4	5	6	7
$x > 0$	0	1	0	0	1	1	0	1
$y > 0$	0	0	1	0	1	0	1	1
$z > 0$	0	0	0	1	0	1	1	1

Remark: Though the cell type is not required in developing the representation of 3-rectangulations, its inclusion here may be justified on the grounds of completeness (since it is a grid parameter) and because of its particular relevance to the enumeration algorithms discussed in the subsequent sections.

A $[p, 3]$-rectangulation R may be represented by a *proper colouring* of R by allocating p distinct colours to the component cuboids in such a manner that grid cells in the same cuboid have the same colour. For computational reasons the colours are identified with integers. Every colouring of R, $C(R)$, may be represented by the ordered set $\langle C(1), C(2), ..., C(lmn)\rangle$, where $C(q)$ is the colour associated with the grid cell whose Γ-number is q. An important proper colouring is the ϕ-colouring of R, $\phi(R)$, defined as follows. Label the cells, taken in their Γ-number order, with increasing numbers starting with 1 such that grid cells in the same cuboid have the same label: $\phi(R) \equiv \langle \phi(1), \phi(2), ..., \phi(lmn)\rangle$. It is easy to show that:
(1) if $\phi(R_1) = \phi(R_2)$ then $R_1 = R_2$;
(2) $\phi(R) \leqslant_{\mathrm{L}} C(R)$ for any other proper colouring C of R, where $\leqslant_{\mathrm{L}}$ denotes less than or equal to in the lexicographical sense.
Consequently $\phi(R)$ may be chosen as the representation of R.

Recall that 3-rectangulations are equivalent to one another if they are identical after a sequence of 'right-handed' rotations or reflections. Consider the group of transformations that maps a 3-rectangulation onto another in its equivalence class. In order to find a representative for each equivalence class, without loss in generality we may restrict our attention to those 3-rectangulations with $l \geqslant m \geqslant n$ (see definition 6) and thus to those transformations that leave the unit grid invariant in space. Let T denote the group of such transformations; T is isomorphic to a (usually small-order) subgroup of the *even* subgroup of O_3, the orthogonal group of rotations of the cube.

[A group is even if all its elements are even; that is, each element is isomorphic to a composition of an even number of axial inversions (or sign changes) and axial interchanges (or coordinate transpositions). The even elements of O_3 correspond to 'right-handed' rotations or reflections.] T depends upon the relative dimensions of the unit grid. All possible even subgroups of O_3 and the cell-to-cell coordinate-map representations of their elements are listed in Krishnamurti and Roe (1979). The grid-invariant transformations of a 3-rectangulation R are denoted by τR, for each $\tau \in T$.

10. A 3-rectangulation R is *canonical* if and only if $\phi(R) \leqslant_{\mathrm{L}} \phi(\tau R)$ for all $\tau \in T$. [$\phi(R)$ is well-defined since if $\phi(R) = \phi(\tau R)$ then $R = \tau R$, in which case R is said to possess the symmetry of τ. A canonical 3-rectangulation is taken as the representative of the equivalence class of 3-rectangulations under T.]

Unfortunately $\phi(\tau R)$ cannot be directly computed from $\phi(R)$. Instead define

$$\tau\phi(R) \equiv \langle \phi[\tau^{-1}(1)], \phi[\tau^{-1}(2)], ..., \phi[\tau^{-1}(lmn)] \rangle ,$$

where $\tau^{-1}(q)$ is the Γ-number of the cell that maps to (x_q, y_q, z_q) under $\tau \in T$. Let p be the content of R. Then there exists a permutation of the numbers from 1 through p such that $\pi\tau\phi(R) = \phi(\tau R)$. The permutation π is constructed as follows.

For all i mark $\pi(i)$ as 'undefined'. Mark the numbers from 1 through p as 'unused'. Select cells q from $\langle 1, 2, ..., lmn \rangle$ in order. Let $q' = \tau^{-1}(q)$. Two cases arise.

Case 1: $\pi[\phi(q')]$ is undefined. Let u be the smallest unused number. Mark u as 'used'. Assign u as the value of $\pi[\phi(q')]$ and as the new colour of q.

Case 2: $\pi[\phi(q')]$ is defined. Assign this number as the new colour of q.

It is clear that π is a permutation. Since the new colours are associated with the cells in increasing order, it follows that $\pi\tau\phi(R)$ is a ϕ-colouring of the transformed 3-rectangulation τR: that is, $\pi\tau\phi(R) = \phi(\tau R)$.

Standard and trivalent 3-rectangulations

The enumeration algorithm for 3-rectangulations may be described by the following recursive backtrack procedure. Construct all possible q-tuples of integers from the current $(q-1)$-tuple of integers until $q = lmn$, with the condition that the resulting lmn-tuple is a ϕ-colouring of some 3-rectangulation. The construction begins with the 1-tuple $\langle 1 \rangle$.

Since each stage q of this procedure corresponds to determining $\phi(q)$, the colour associated with the cell whose Γ-number is q, given $\langle \phi(1), \phi(2), ..., \phi(q-1) \rangle$, the construction may be recast in terms of 'colouring rules'.

The colouring rules may be expressed in terms of a colouring grammar, which consists of a set of grid parameters, a set of predicates, a set of colours, a set of colouring rules, and an initial colouring rule, where the rules take the form

$$\text{uncoloured shape} \xrightarrow{\text{predicate}} S_1 ; \text{ coloured shape}; \ S_2 .$$

The *uncoloured shape* is a collection of grid cells with at least one uncoloured cell, that results in the *coloured shape* after the application of the rule. S_1 and S_2 are sets of assignations involving grid parameters before and after the rule has been applied. The colouring rules depend upon the relative relationship between coloured cells in the uncoloured shape, and hence may be expressed in terms of parametrized colours, with the actual substitution taking place when the rule is applied. The *predicate* is a Boolean-valued expression which must be satisfied in order that the entire rule may apply.

The rule application may be described as follows. Let q denote the current uncoloured cell in the threading pattern Γ. Select the uncoloured shape that corresponds to the neighbourhood of q. Then if the predicate, which is usually expressed as a functional in q, holds, apply the rule and colour cell q accordingly.

(More than one uncoloured cell may be coloured at any given stage, in which case the 'next' q in Γ is explicitly assigned in one of the two sets of assignations S_1 and S_2; otherwise the 'next' q points to the next uncoloured cell in Γ and by convention no explicit assignment of q is stated in the rules.) A colouring grammar is a complete recursive specification for 3-rectangulations.

It is convenient to have a pictorial description of the colour and spatial relationships between the grid cells. Consider the graph representation in which edges correspond to adjacent pairs of grid cells, and bold edges pairs of grid cells in the same cuboid. Each grid cell is represented by an open vertex which is darkened when the cell is coloured. A vertex may be labelled by the colour (or parametrized colour) of the grid cell it represents. In order to simplify the colouring rules a third type of edge, represented by dotted lines, which may stand for either a single edge or a bold edge, is introduced. Figure 2 presents the graph of a [5, 3]-rectangulation on the (2, 2, 2) unit grid.

Since the colouring rules act according to some local criteria, it is necessary that an enumerative grammar accounts for all possible uncoloured shapes that can occur. One possibility is to view the uncoloured shape as the neighbourhood of grid cells that surround the intersections of grid planes. These intersections of the grid planes have been previously classified in terms of cell types (see definition 9). Let q denote the current uncoloured cell in Γ. Let r, s, t, q', r', s', and t' denote the parametrized

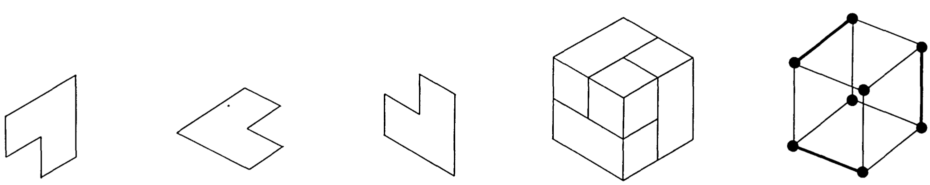
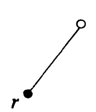
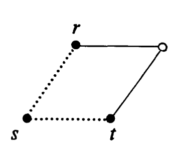
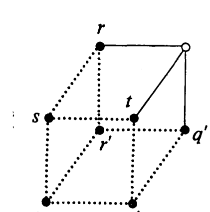


Figure 2. A [5, 3]-rectangulation on the (2,2,2) unit grid, and its graph representation.

Table 1. All possible uncoloured shapes in a 3-rectangulation.

Cell type	r, s, t, etc	Uncoloured shape
0		o
1	$r = \phi(q-1)$	
2	$r = \phi(q-l)$	
3	$r = \phi(q-lm)$	
4	$r = \phi(q-1)$ $s = \phi(q-l-1)$ $t = \phi(q-l)$	
5	$r = \phi(q-1)$ $s = \phi(q-lm-1)$ $t = \phi(q-lm)$	
6	$r = \phi(q-l)$ $s = \phi(q-lm-l)$ $t = \phi(q-lm)$	
7	$r = \phi(q-1)$ $s = \phi(q-l-1)$ $t = \phi(q-l)$ $q' = \phi(q-lm)$ $r' = \phi(q-lm-1)$ $s' = \phi(q-lm-l-1)$ $t' = \phi(q-lm-l)$	

colours of the cells neighbouring q. The uncoloured shapes then take the forms indicated in table 1 (note that the open vertex represents q).

The colouring rules may be easily determined using the information in table 1. One condition that rules must ensure is that eventually only cuboids are created. A desirable condition is that each 3-rectangulation is generated by a unique sequence of rule applications.

Let μ denote the number of colours used so far.

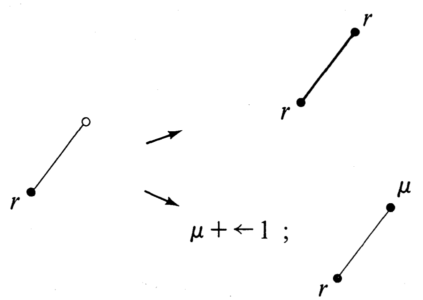
Case 1: Type = 0.
The grid cell is the first cell in Γ. By the ϕ-colouring requirements it must be coloured 1. The colouring rule is therefore

$$0\colon\ \circ \longrightarrow \bullet^{1};\ \mu \leftarrow 1\ .$$

Rule 0 may be considered as the initial colouring rule.

Case 2: Type = 1, 2, or 3.
Here q may take on one of two colours: either r or a 'new' colour. That is, q is in the same cuboid as its adjacent coloured neighbour or in a different cuboid. The colouring rules are as follows:

1–2:



Case 3: Type = 4, 5, or 6.
There are four possible uncoloured shapes that can arise; these are illustrated in figure 3(a). For each situation the colouring rules that may apply are those in figure 3(b). These rules can be combined and reduced to a smaller set by means of predicates. Consider the predicates

$$G_1\colon\ (r = t) \vee (r \neq s)\ , \qquad G_2\colon\ (r \neq t) \wedge (s \neq t)\ , \quad \text{and} \quad G_3\colon\ r \neq t\ .$$

Then the eight rules in figure 3(b) can be replaced by the three rules in figure 3(c).

Case 4: Type = 7.
In this case there are sixty-seven possible uncoloured shapes that can occur, which result in one hundred and thirty-five colouring rules. A few examples are shown in figure 4(a). The colouring rules can be reduced to a set of four rules. The reason that only four rules are sufficient stems from the fact that cell q can either take on one of the colours of its three adjacent coloured neighbours or be assigned a new colour. The four rules are shown in figure 4(b), where the predicates are given by

$$G_4\colon\ (r = t) \vee (r = q') \vee [(r \neq r') \wedge (r \neq s) \wedge (t \neq q')]\ ,$$

$$G_5\colon\ (r \neq t) \wedge (r \neq q') \wedge \{(t = q') \vee [(s \neq t) \wedge (t \neq t')]\}\ ,$$

$$G_6\colon\ (r \neq t) \wedge (r \neq q') \wedge (t \neq q') \wedge (q' \neq r') \wedge (q' \neq t')\ ,$$

$$G_7\colon\ (r \neq t) \wedge (r \neq q') \wedge (t \neq q')\ .$$

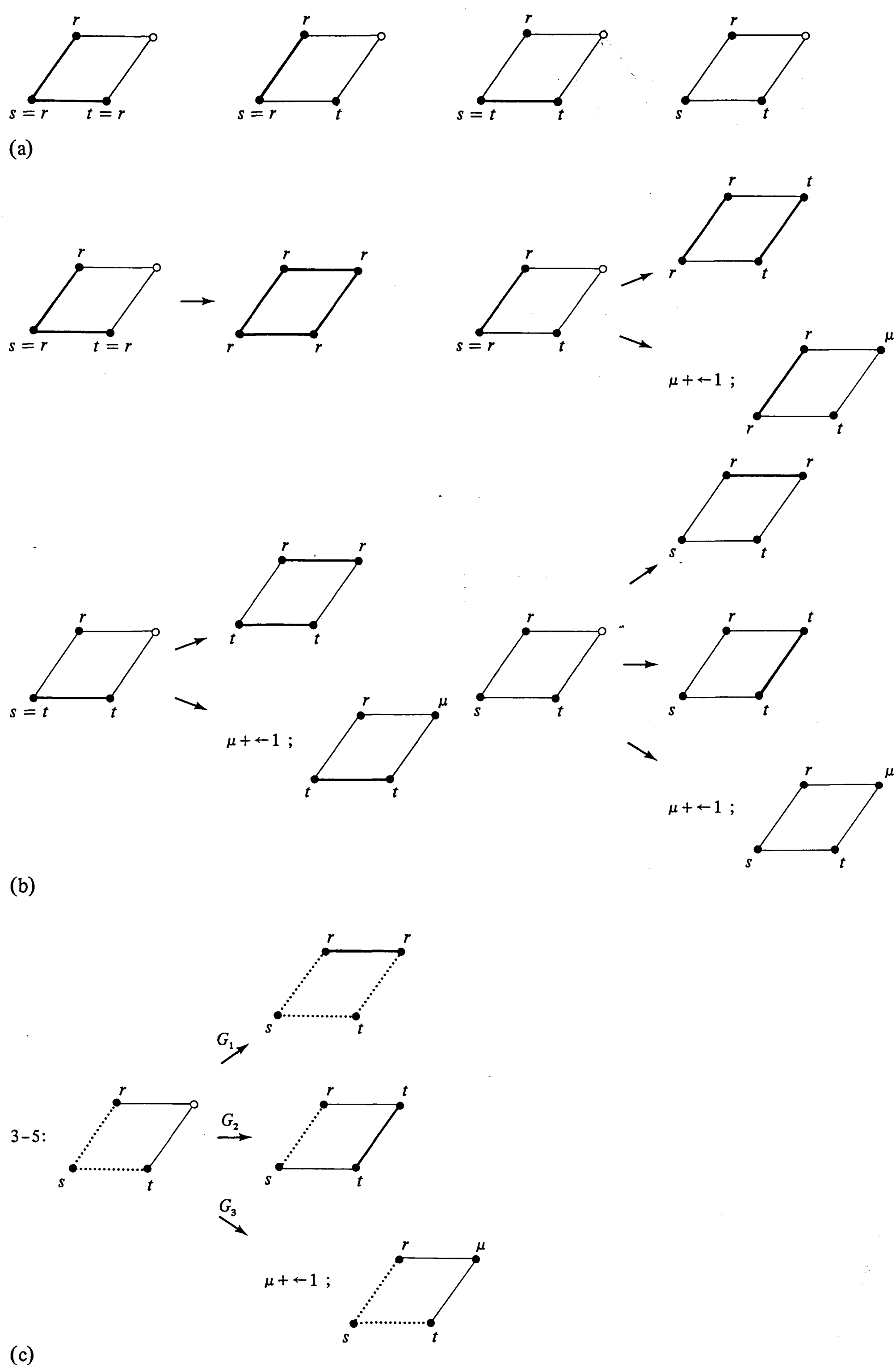


Figure 3. The colouring rules for cells of type 4, 5, or 6.

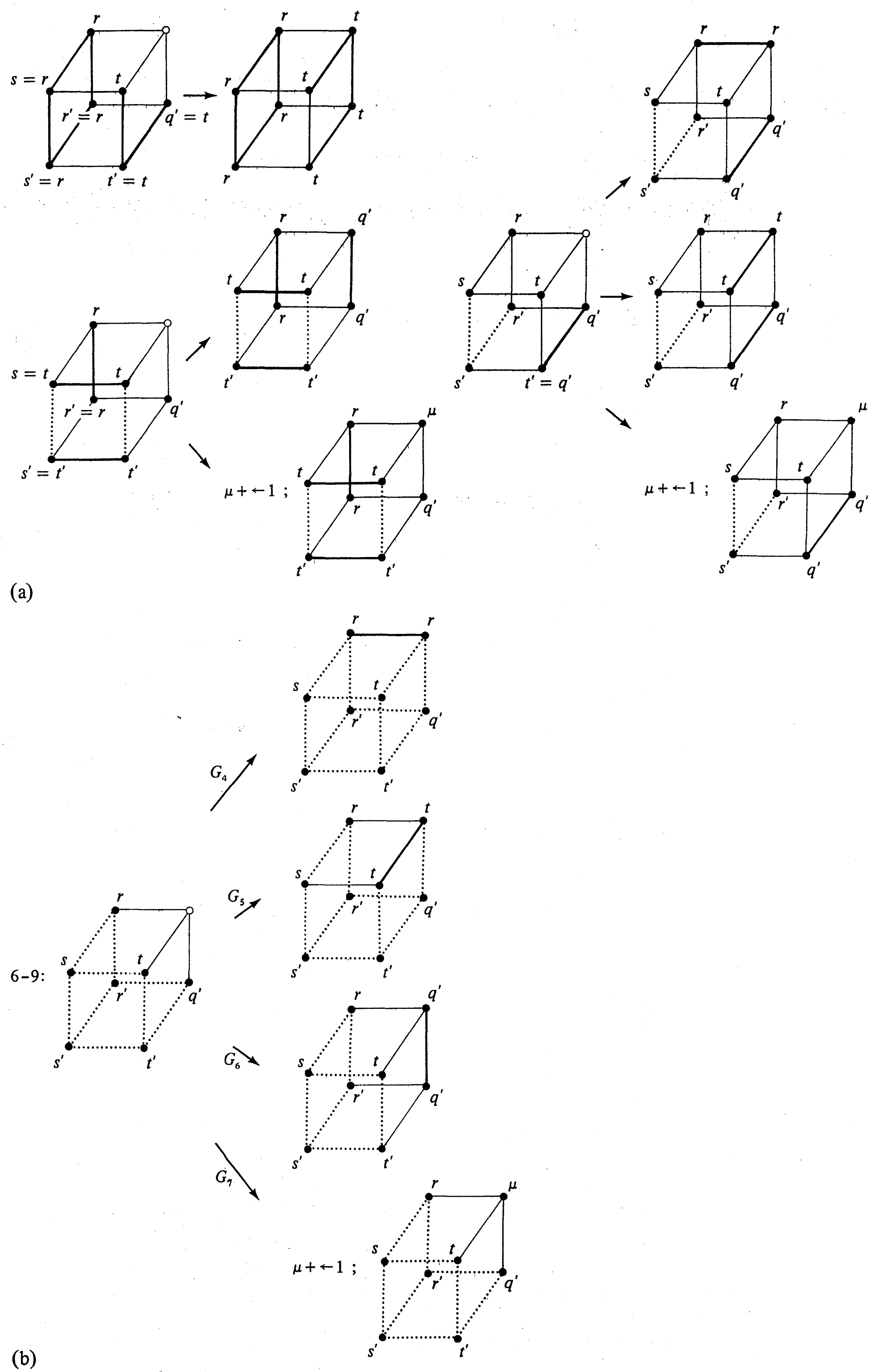


Figure 4. The colouring rules for cells of type 7.

q: current cell in threading pattern Γ
γ_q: cell type of q

Note: $\bar{F}$ denotes the complement of F.

Predicates

F_1: $1 \leqslant \gamma_q \leqslant 3$
F_2: $4 \leqslant \gamma_q \leqslant 6$
F_3: $\gamma_q = 7$
F_4: $lmn - q \geqslant p - \mu$
F_5: $p > \mu$
F_6: $(x_q = l-1) \wedge (y_q = m-1)$
F_7: $(y_q = m-1) \wedge (z_q = n-1)$
F_8: $(z_q = n-1) \wedge (x_q = l-1)$
F_9: $\{[F_6 \wedge (h_{z_q} > 0)] \vee \bar{F}_6\} \wedge \{[F_7 \wedge (f_{x_q} > 0)] \vee \bar{F}_7\} \wedge \{[F_8 \wedge (g_{y_q} > 0)] \vee \bar{F}_8]\}$
F_{10}: $(r = t) \vee (r \neq s)$
F_{11}: $(r \neq t) \wedge (s \neq t)$
F_{12}: $r \neq t$
F_{13}: $(r = t) \vee (r = q') \vee [(r \neq r') \wedge (r \neq s) \wedge (t \neq q')]$
F_{14}: $(r \neq t) \wedge (r \neq q') \wedge \{(t = q') \vee [(s \neq t) \wedge (t \neq t')]\}$
F_{15}: $(r \neq t) \wedge (r \neq q') \wedge (t \neq q') \wedge (q' \neq r') \wedge (q' \neq t')$
F_{16}: $(r \neq t) \wedge (r \neq q') \wedge (t \neq q')$

P_1: $F_1 \wedge F_4$
P_2: $F_1 \wedge F_5$
P_3: $F_2 \wedge F_4 \wedge F_{10}$
P_4: $F_2 \wedge F_4 \wedge F_{11}$
P_5: $F_2 \wedge F_5 \wedge F_{12}$
P_6: $F_3 \wedge F_4 \wedge F_q \wedge F_{13}$
P_7: $F_3 \wedge F_4 \wedge F_9 \wedge F_{14}$
P_8: $F_3 \wedge F_4 \wedge F_9 \wedge F_{15}$
P_9: $F_3 \wedge F_5 \wedge F_{16}$

Assignations

S_1: $(x_q > 0 \Rightarrow f_{x_q} \leftarrow 1) \vee (y_q > 0 \Rightarrow g_{y_q} \leftarrow 1) \vee (z_q > 0 \Rightarrow h_{z_q} \leftarrow 1)$
S_2: $(x_q = 0 \Rightarrow g_{y_q} + \leftarrow 1, h_{z_q} + \leftarrow 1) \vee (y_q = 0 \Rightarrow h_{z_q} + \leftarrow 1, f_{x_q} + \leftarrow 1) \vee (z_q = 0 \Rightarrow f_{x_q} + \leftarrow 1, g_{y_q} + \leftarrow 1)$
S_3: $f_{x_q} + \leftarrow 1, g_{y_q} + \leftarrow 1, h_{z_q} + \leftarrow 1$

Rules

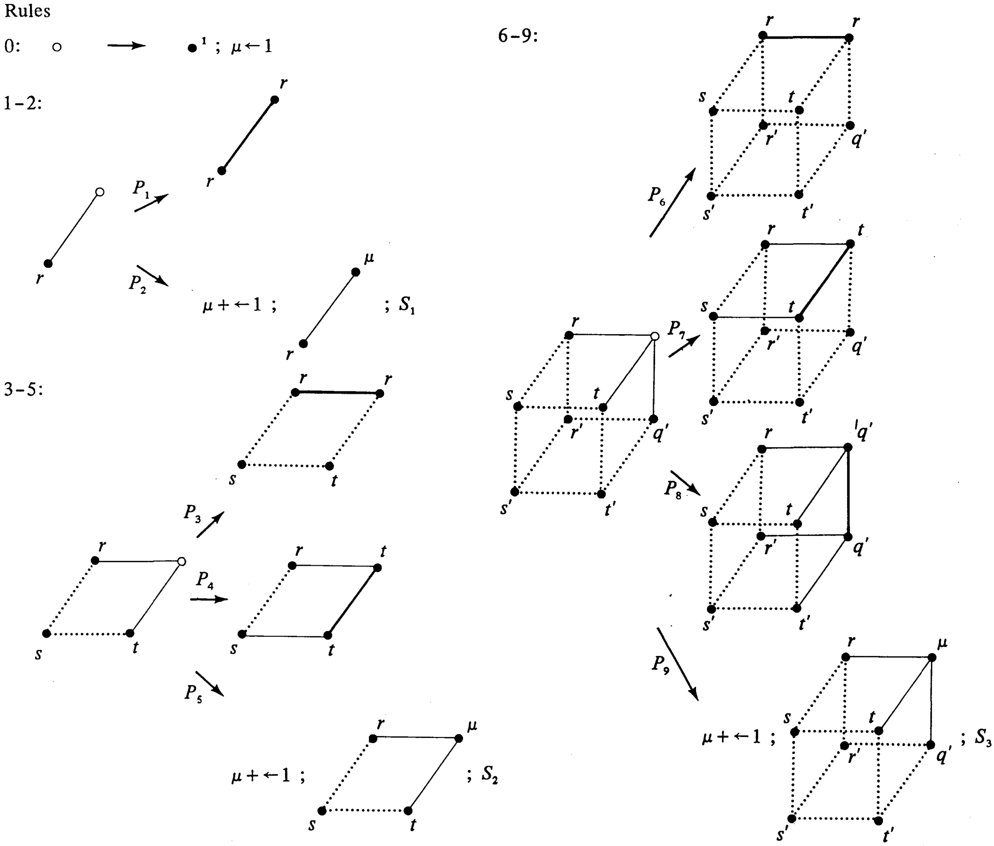


Figure 5. The colouring grammar which generates all proper standard $[p, 3]$-rectangulations, for p fixed.

We still have a fair way to go. Rules 0 through 9 will only generate 3-rectangulations without any consideration as to whether the generated object is standard, trivalent, or has content p. Suppose we wish to generate standard $[p, 3]$-rectangulations, for p fixed. In such cases we must ensure that (a) μ never exceeds p and (b) each grid plane contains at least one face. This can be accomplished by incorporating Boolean expressions into the colouring rules. Let $\{f_x\}$, $\{g_y\}$, and $\{h_z\}$ be sets of integer variables that denote the absence or presence of a maximal plane in the corresponding grid planes. That is, $f_{x_0} > 0$ if there exists at least one face in the grid plane $X = x_0$, and so on. Then the colouring grammar in figure 5 will generate, by unique sequences of rule applications, all proper standard $[p, 3]$-rectangulations, for p fixed. To ensure that the 3-rectangulations are always trivalent, rules 5 and 9 (in figure 5) must be replaced by rules $5'$ and $9'$ shown in figure 6. The colouring grammar in figure 5, modified by the rules in figure 6, will generate, again by unique sequences of rule applications, all proper standard trivalent $[p, 3]$-rectangulations, for p fixed.

F_{17}: $(r = s) \vee (s = t)$
F_{18}: $[(r = s) \vee (s = t)] \wedge [(t = t') \vee (t' = q')]$
$\qquad \wedge\ [(r = r') \vee (r' = q')]$
$P_{5'}$: $F_2 \wedge F_5 \wedge F_{12} \wedge F_{17}$
$P_{9'}$: $F_3 \wedge F_5 \wedge F_{16} \wedge F_{18}$

5':

9':

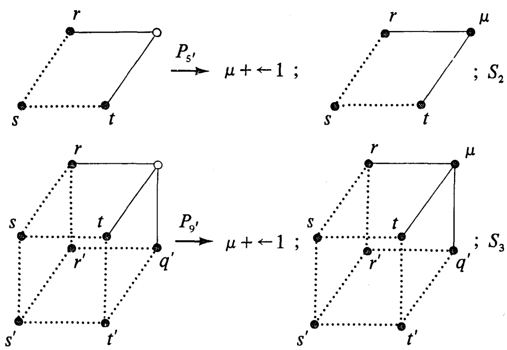


Figure 6. The modified rules 5 and 9 required to generate all proper standard trivalent $[p, 3]$-rectangulations, for p fixed.

Nonaligned 3-rectangulations

The following procedure describes a sieve for detecting nonaligned 3-rectangulations. Let δ denote any internal grid plane of a 3-rectangulation. For instance δ may be identical to $X = x \in \mathrm{Z}_l^+$, or identical to $Y = y \in \mathrm{Z}_m^+$, or identical to $Z = z \in \mathrm{Z}_n^+$. The maximal planes in δ have a graph representation in which edges represent pairs of adjacent grid faces which are in the same maximal plane. (Two grid faces in a plane are *adjacent* if they share either a line or a point.) The resulting graph is termed the δ-graph of the 3-rectangulation. A 3-rectangulation is nonaligned if and only if each of its δ-graphs is either null or connected. Algorithmically connectedness is an efficient graph property (Tarjan, 1972). Thus, given a family of 3-rectangulations, its subfamily of nonaligned 3-rectangulations may be easily determined.

From a computational standpoint this two-step generate-and-test algorithm can be improved upon. The global criterion to decide on the nonalignment of a 3-rectangulation may be incorporated into the colouring grammar for 3-rectangulations as sets of local rules in such a manner that the resulting colouring grammar will generate precisely the family of nonaligned 3-rectangulations.

The local rules may be described by the following tagging procedure. A 'tag' is assigned to each grid face in every internal grid plane so that grid faces in the same maximal plane have the same tag. Distinct maximal planes in the same internal grid plane have distinct tags. By convention grid faces in no maximal plane have the 'null tag'. The tag associated with a grid plane is the largest tag, in the numerical sense, amongst those of all the maximal planes it contains. Clearly a 3-rectangulation is nonaligned if and only if the tag of each of its grid planes does not exceed unity.

In order that this tagging procedure may be successfully carried out, the tags must satisfy the following conditions.

11. A map α: $Z^{\oplus} \to Z^{\oplus}$ is a *tag function* if and only if each $n \in Z^{\oplus}$ satisfies the following:
(a) $\alpha(n) \leqslant n$;
(b) $\alpha(n > 0) > 0$;
(c) there exists a t_n such that $\alpha^{t_n}(n) \leqslant \alpha(n)$ and $\alpha^{t_n+1}(n) = \alpha^{t_n}(n)$.
For each $n \in Z^{\oplus}$, $\alpha^{t_n}(n) = \alpha^{t_n+1}(n) = ... = \alpha^*(n)$ is called the *tag* of n. ($Z^{\oplus}$ is the set of nonnegative integers.)
[The least t_n will vary according to n, but there must exist an $\alpha^*(n)$ for all $n \in Z^{\oplus}$.]

A general description of the tagging procedure is now presented. Let Σ_δ denote the set of grid faces in the grid plane δ. To each $\sigma \in \Sigma_\delta$ may be assigned a label which is the Γ-number of the grid cell whose coordinates it shares. That is, if σ has coordinate $(x_\sigma, y_\sigma, z_\sigma)$, σ is assigned $\Gamma(x_\sigma, y_\sigma, z_\sigma)$. For convenience let this number be referred to as $\Gamma(\sigma)$. Let $A(\sigma)$ denote the following adjacency set of σ:

$$A(\sigma) = \{\rho \in \Sigma_\delta \colon \rho \text{ and } \sigma \text{ are adjacent and } \Gamma(\rho) < \Gamma(\sigma)\}\,.$$

Mark the numbers $1, 2, ..., |\Sigma_\delta|$ as 'untagged'. Set $\alpha(0) = 0$. Select faces $\sigma \in \Sigma_\delta$ in increasing order of their Γ-numbers and assign to each in turn the tag label $\beta(\sigma)$. Two possible cases arise.

Case 1: σ is not contained in any maximal plane.
Assign zero to $\beta(\sigma)$.

Case 2: σ is contained in some maximal plane.
Let $A'(\sigma) = \{\rho \in A(\sigma)\colon \beta(\rho) > 0\}$. Two cases arise.
Case 2.1: $A'(\sigma) = \emptyset$.
Let u be the smallest untagged number. Assign u to $\beta(\sigma)$. Tag u; that is, set $\alpha(u) = u$.
Case 2.2: $A'(\sigma) \neq \emptyset$.
Let v denote the minimum $\alpha^*[\beta(\rho)]$ over all $\rho \in A'(\sigma)$. Assign v to $\beta(\sigma)$. Retag $A'(\sigma)$; that is, for each $\rho \in A'(\sigma)$ set $\alpha[\beta(\rho)] = \alpha^2[\beta(\rho)] = ... = \alpha^*[\beta(\rho)] = v$.

Clearly α is a tag function, and $\{\alpha^*[\beta(\sigma)]\colon \sigma \in \Sigma_\delta\}$ is the set of tags of the grid faces in δ. Moreover it is easy to show that in general

$$\max_{\sigma \in \Sigma_\delta} \{\alpha^*[\beta(\sigma)]\} \geqslant \text{number of maximal planes in } \delta.$$

Equality always holds whenever the number of maximal planes is either zero or one. A 3-rectangulation is nonaligned if and only if for each internal grid plane δ

$$\max_{\sigma \in \Sigma_\delta} \{\alpha^*[\beta(\sigma)]\} \leqslant 1\ .$$

The remainder of this section is devoted to describing the tagging rules that are incorporated into the colouring rules for 3-rectangulations in order that only nonaligned 3-rectangulations are generated. Let f, g, and h be the prefix symbols for variables involving the constant-x, constant-y, and constant-z grid planes respectively. Let $\{f_x\}$, $\{g_y\}$, and $\{h_z\}$ denote the numbers of maximal planes on each grid plane, all initially set equal to zero. Let $\{f\beta_q\}$, $\{g\beta_q\}$, and $\{h\beta_q\}$ denote the β-values of the grid faces, which are also initially set equal to zero. Let $\{f\mu_x\}$, $\{g\mu_y\}$, and $\{h\mu_z\}$ denote the current largest tagged numbers in the appropriate grid planes. Let $\{f\alpha_x\}$, $\{g\alpha_y\}$, and $\{h\alpha_z\}$ represent the tag functions. Initially $f\alpha_x(0) = 0$ for all $x \in Z_l^+$, $g\alpha_y(0) = 0$ for all $y \in Z_m^+$, and $h\alpha_z(0) = 0$ for all $z \in Z_n^+$. Let q denote the current cell in Γ; q also denotes the current grid face to be tagged. Since a grid face is contained in a maximal plane if and only if the grid cells that share the face are coloured differently, we need only consider those colouring rules in which cell q is coloured differently from at least one of its coloured neighbours. The following cases have to be considered.

Case 1: Cell type = 1, 2, or 3.
Here only rule 2 needs to be considered; that is, when $\phi(q) \neq r$. The grid face separating q and its coloured neighbour has no adjacent tagged faces. It is in fact the first face encountered in the appropriate grid plane. Consequently it is the first recorded instance of a maximal plane in that grid plane. Which grid plane is involved depends upon the cell type of q. The tagging rule takes the form

$$j_{w_q} \leftarrow j\beta_q \leftarrow j\mu_{w_q} \leftarrow j\alpha_{w_q}(1) \leftarrow 1 \ ,$$

where (j, w) stands for (f, x), (g, y), or (h, z) depending on whether the cell type is 1, 2, or 3 respectively.

Case 2: Cell type = 4, 5, or 6.
Except when $r = t$ each colouring rule requires at least one grid face to be tagged (see rules 3, 4, and 5). Let γ denote the cell type and let $\langle A_\gamma, \delta, j, w\rangle$ represent the adjacency set of the grid face q in the grid plane δ (j and w have the same meaning as in case 1). Two cases have to be considered.

Case 2.1: $\phi(q) \neq r$.
Here

$$\langle A_\gamma, \delta, j, w\rangle = \begin{cases} \langle\{q-l\}, X = x_q, f, x\rangle & \text{if } \gamma = 4 \ , \\ \langle\{q-lm, q-lm+l\}, X = x_q, f, x\rangle & \text{if } \gamma = 5 \ , \\ \langle\{q-lm, q-lm+1\}, Y = y_q, g, y\rangle & \text{if } \gamma = 6 \ . \end{cases}$$

For A_5 and A_6 the two grid cells either have the same tag, which may be zero, or one has a zero and the other a nonzero tag. This assertion may be verified when all the tagging rules have been developed. Let $A_4' = \{\sigma \in A_4 : f\beta_\sigma > 0\}$, $A_5' = \{\sigma \in A_5 : f\beta_\sigma > 0\}$, and $A_6' = \{\sigma \in A_6 : g\beta_\sigma > 0\}$. Two cases have to be considered.

Case 2.1.1: $A_\gamma' = \emptyset$.
In this case the grid face represents the first face in a possible new maximal plane in the grid plane δ. The tag of face q is set to the current smallest untagged number and the tag function for this tag is set to map onto itself. The tagging rule is

$$j_{w_q} + \leftarrow 1 \ ,$$

$$j\alpha_{w_q}[j\beta_q \leftarrow (j\mu_{w_q} + \leftarrow 1)] \leftarrow j\beta_q \ .$$

Case 2.1.2: $A_\gamma' \neq \emptyset$.
In this case face q is adjacent to a face which is tagged and therefore must be contained in the same maximal plane as its neighbour. Let σ be a face in A_γ'. The tagging rule is

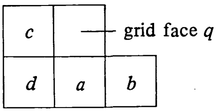
$$j\beta_q \leftarrow j\beta_\sigma \ .$$

Case 2.2: $\phi(q) \neq t$.
Here

$$\langle A_\gamma, \delta, j, w\rangle = \begin{cases} \langle\{q-1\}, Y = y_q, g, y\rangle & \text{if } \gamma = 4 \ , \\ \langle\{q-1\}, Z = z_q, h, z\rangle & \text{if } \gamma = 5 \ , \\ \langle\{q-l, q-l+1\}, Z = z_q, h, z\rangle & \text{if } \gamma = 6 \ . \end{cases}$$

As in case 2.1, A'_γ may be defined for each $\gamma = 4, 5$, and 6. The two resulting subcases, for $A'_\gamma = \emptyset$ and $A'_\gamma \neq \emptyset$, may then be treated in the same manner as were the corresponding subcases of case 2.1. In each case the tagging rule takes the identical form as previously given.

Case 3: Cell type = 7.
Consider the grid plane $X = x_q$ separating the sets of cells $\{q, q-l, q-lm, q-lm-l\}$ and $\{q-1, q-l-1, q-lm-1, q-lm-l-1\}$ in the uncoloured shape in rules 6 through 9. Suppose cell q is coloured differently from cell $q-1$. In this situation grid face q in the plane $X = x_q$ has to be tagged. All the faces adjacent to q and which are already tagged are as follows.



The letters a, b, c, and d refer to the tags of the neighbouring grid faces. They are given by

$$a = f\beta_{q-lm} \,,$$

$$b = \begin{cases} 0 & \text{if } y_q = m-1 \,, \\ f\alpha^*_{x_q}(f\beta_{q-lm+l}) & \text{otherwise,} \end{cases}$$

$$c = f\alpha^*_{x_q}(f\beta_{q-l}) \,,$$

$$d = f\alpha^*_{x_q}(f\beta_{q-lm-l}) \,.$$

There are two cases to be considered.

Case 3.1: $a \neq 0$.
In this case the tagging rule is simply

$$f\beta_q \leftarrow a \,.$$

Case 3.2: $a = 0$.
There are four subcases to be considered.
Case 3.2.1: $b = c = d = 0$.
As in case 2.1.1 a possible new maximal plane is introduced. The tagging rule is

$$f_{x_q} + \leftarrow 1 \,,$$

$$f\alpha_{x_q}[f\beta_q \leftarrow (f\mu_{x_q} + \leftarrow 1)] \leftarrow f\beta_q \,.$$

Case 3.2.2: $b = 0$ and at least one of c and d is nonzero.
Here face q is tagged with the same tag as either face $q-l$ or face $q-lm-l$. (If they are both tagged they must have the same tag.) Therefore the tagging rule is

$$f\beta_q \leftarrow \max\{c, d\}$$

Case 3.2.3: $b \neq 0$ and $c = d = 0$.
This is similar to case 3.2.2 and the tagging rule is

$$f\beta_q \leftarrow b \,.$$

Case 3.2.4: $b \neq 0$ and at least one of c and d is nonzero.
Either $b = \max\{c, d\}$, in which case the grid faces are in the same maximal plane and this situation may be treated as a special case of either cases 3.2.2 or 3.2.3, or

$b \neq \max\{c, d\}$. In the latter situation the tagging of face q involves combining two distinct maximal planes into a single plane and retagging one of the planes. The tagging rule is

$$f_{x_q} - \leftarrow 1 \ ,$$
$$f\beta_q \leftarrow \min\left\{b, \max\{c, d\}\right\} \ ,$$
$$f\alpha^*_{x_q}\left(\max\left\{b, \max\{c, d\}\right\}\right) \leftarrow f\beta_q \ .$$

The grid faces in the planes $Y = y_q$ and $Z = z_q$ are identically handled. So, to generate nonaligned 3-rectangulations, in the case of cells of type 7 the colouring rules must also be controlled by predicates which ensure that all the grid planes have no more than one maximal plane. Suppose face q in the plane $X = x_q$ has the coordinates $(x_q, m-1, n-1)$. This face is the last face in $X = x_q$ to be tagged. Consequently the colouring rule for cell $q+1$ may be applied only if $f_{x_q} \leqslant 1$. The rules are similarly controlled when the last face q in the $Y = y_q$ and $Z = z_q$ planes is encountered.

Algorithms

Three algorithms in ALGOL-like notation which form the nucleus of the enumeration algorithm are presented in this section.

Algorithm 1 is a Boolean-valued procedure for determining whether the given 3-rectangulation, R, is canonical in the sense of definition 10. Briefly the algorithm may be described as follows. The outer loop selects a τ in T and proceeds to test whether $\phi(R) \leqslant_{\mathrm{L}} \phi(\tau R)$ providing *flag* remains **true**. The variable *flag* signifies whether R is canonical with respect to the previous set of transformations that have been applied, and is initially **true**. The inner loop selects cells q in order and applies the rules for the construction of the permutation π. The loop is continued as long as $\phi(R)$ and $\phi(\tau R)$ correspond term by term. The last statement in the outer loop is the actual test [when $\phi(R) \neq \phi(\tau R)$] which determines whether R is canonical with respect to τ. The algorithm returns the value **true** if and only if R is canonical.

Algorithm 2 gives the rules for tagging the grid faces after cell q has been coloured. This routine is performed in conjunction with the colouring rules (see algorithm 3), and consequently any changes to the grid variables must be recorded for purpose of backtracking. Three sets, B_q, U_q, and V_q, are employed for this reason. B_q contains elements of the form (j, w), which signify that at construction level q the variables $j\beta_q$ and $j\alpha_{w_q}(j\beta_q)$ have been introduced and the variables j_{w_q} and $j\mu_{w_q}$ have been incremented by unity. In other words at level q a new maximal plane has been

Algorithm CANONICAL(R)
¶ R and π are represented by the ordered set $\langle\phi(1), \phi(2), ..., \phi(lmn)\rangle$ and the array $\pi[1, ..., p]$ respectively ¶
flag $\leftarrow$ **true**
for $\tau \in T$ **while** *flag* **do**
  for $i \in \langle 1, 2, ..., p\rangle$ **do** $\pi(i) \leftarrow$ 'undefined'
  unused number $\leftarrow 0$
  for $q \in \langle 1, 2, ..., lmn\rangle$ **while** *flag* **do**
    if $\pi[\phi(\tau^{-1}(q))]$ is undefined
    then *newcolour* $\leftarrow \pi[\phi(\tau^{-1}(q))] \leftarrow$ (*unused number* $+ \leftarrow 1$)
    else *newcolour* $\leftarrow \pi[\phi(\tau^{-1}(q))]$
    flag $\leftarrow$ *newcolour* $= \phi(q)$
  if not *flag* **then** *flag* $\leftarrow \phi(q) <$ *newcolour*
return (*flag*)

Algorithm 1.

Algorithm TAGRULE

```
¶ For purposes of backtracking, changes to the grid planes, the tag functions, and the number of
  maximal planes are recorded in the sets B_q, V_q, and U_q respectively ¶
B_q ← V_q ← U_q ← ∅
nonalignment ← true
if q > 1
then
    if celltype(q) ≤ 3
    then
        (j, w) ← case celltype(q) in
                    (= 1) (f, x)
                    (= 2) (g, y)
                    (= 3) (h, z)
        j_{w_q} ← jβ_q ← jμ_{w_q} ← jα_{w_q}(1) ← 1
        B_q ← {(j, w)}
    else
        if celltype(q) ≤ 6
        then
            ⟨r, s, t⟩ ← case celltype(q) in
                    (= 4) ⟨q−1, q−l−1, q−l⟩
                    (= 5) ⟨q−1, q−lm−1, q−lm⟩
                    (= 6) ⟨q−l, q−lm−l, q−lm⟩
            for θ ∈ {r, t} do
                if φ(q) ≠ φ(θ)
                then
                    ⟨j, w, β⟩ ← case celltype(q) in
                        (= 4) if θ = r then ⟨f, x, fβ_t⟩ else ⟨g, y, gβ_r⟩
                        (= 5) if θ = r then ⟨f, x, max{fβ_t, fβ_{t+l}}⟩ else ⟨h, z, hβ_r⟩
                        (= 6) if θ = r then ⟨g, y, max{gβ_t, gβ_{t+1}}⟩ else ⟨h, z, max{hβ_r, hβ_{r+1}}⟩
                    if β ≠ 0
                    then jβ_q ← β
                    else
                        j_{w_q} +← 1
                        jα_{w_q}[jβ_q ← (jμ_{w_q} +← 1)] ← jβ_q
                        B_q ← B_q ∪ {(j, w)}
        else
            ⟨r, t, q'⟩ ← ⟨q−1, q−l, q−lm⟩
            for ⟨j, w, θ, β⟩ ∈ {⟨f, x, r, fβ_{q'}⟩, ⟨g, y, t, gβ_{q'}⟩, ⟨h, z, q', hβ_t⟩} do
                if φ(q) ≠ φ(θ)
                then
                    if β ≠ 0
                    then jβ_q ← β
                    else
                        ⟨b, c, d⟩ ← case j in
                            (= f) ⟨if y_q = m−1 then 0 else fβ_{q'+l}, fβ_t, fβ_{q'−l}⟩
                            (= g) ⟨if x_q = l−1 then 0 else gβ_{q'+1}, gβ_r, gβ_{q'−1}⟩
                            (= h) ⟨if x_q = l−1 then 0 else hβ_{t+1}, hβ_r, hβ_{t−1}⟩
                        if b+c+d = 0
                        then
                            j_{w_q} +← 1
                            jα_{w_q}[jβ_q ← (jμ_{w_q} +← 1)] ← jβ_q
                            B_q ← B_q ∪ {(j, w)}
                        else
                            if b = 0
                            then jβ_q ← max{c, d}
                            else
                                if c+d = 0
                                then jβ_q ← b
                                else
                                    ¶ jα*_{w_q}( ) may be regarded as the following procedure:
                                      let v be such that
                                      k = jα^0_{w_q}(k) > jα^1_{w_q}(k) > ... > jα^v_{w_q}(k) = jα^{v+1}_{w_q}(k);
                                      set V_q to contain the elements
                                      ⟨jα^i_{w_q}(k), jα^{i+1}_{w_q}(k), (j, w)⟩, 0 ≤ i < v;
                                      assign jα^i_{w_q}(k) ← jα^v_{w_q}(k), 1 ≤ i < v;
                                      jα*_{w_q}(k) ≡ jα^v_{w_q}(k) ¶
                                    b ← jα*_{w_q}(b)
                                    c ← jα*_{w_q}(c)
                                    d ← jα*_{w_q}(d)
                                    if b = max{c, d}
                                    then jβ_q ← b
                                    else
                                        jβ_q ← min{b, max{c, d}}
                                        β' ← max{b, max{c, d}}
                                        V_q ← V_q ∪ {⟨β', β', (j, w)⟩}
                                        jα_{w_q}(β') ← jβ_q
                                        j_{w_q} −← 1
                                        U_q ← U_q ∪ {(j, w)}
            ¶ test if the 3-rectangulation is nonaligned ¶
            nonalignment ← [if (x_q = l−1) ∧ (y_q = m−1) then h_{z_q} = 1 else true] and
                           [if (y_q = m−1) ∧ (z_q = n−1) then f_{x_q} = 1 else true] and
                           [if (z_q = n−1) ∧ (x_q = l−1) then g_{y_q} = 1 else true]
return (nonalignment)
```

Algorithm 2.

```
Algorithm 3-RECTANGULATION
¶ Recursive procedure to generate various kinds of proper standard [p, 3]-rectangulations, p fixed,
  on the given (l, m, n) unit grid;
  essentially the procedure constructs φ(q+1) given ⟨φ(1), φ(2), ..., φ(q)⟩ by applying the colouring
  rules in figures 5 and 6 ¶
if if nonalignment desired then TAGRULE else true
then
  if q = lmn
  then if CANONICAL (generated 3-rectangulation) then equivalence class +← 1
  else
    ¶ colour q+1 with a 'new' colour if possible ¶
    if p > μ
    then
      if
        if celltype(q+1) ≤ 3
        then true
        else
          if celltype(q+1) ≤ 6
          then
            ⟨r, s, t⟩ ← case celltype(q+1) in
                  (= 4) ⟨φ(q), φ(q−l), φ(q+1−l)⟩
                  (= 5) ⟨φ(q), φ(q−lm), φ(q+1−lm)⟩
                  (= 6) ⟨φ(q+1−l), φ(q+1−lm−l), φ(q+1−lm)⟩
            r ≠ t and [if trivalency desired then (r = s) ∨ (s = t) else true]
          else
            ⟨r, s, t, q', r', t'⟩ ← ⟨φ(q), φ(q−l), φ(q+1−l), φ(q+1−lm), φ(q−lm),
                                                                   φ(q+1−lm−l)⟩
            [(r ≠ t) ∧ (r ≠ q') ∧ (t ≠ q')]
            and
              if trivalency desired
              then [(r = s) ∨ (s = t)] ∧ [(t = t') ∨ (t' = q')] ∧ [(r = r') ∨ (r' = q')]
              else true
      then
        if not nonalignment desired then f_{x_q} > 0 +← 1, g_{y_q} > 0 +← 1, h_{z_q} > 0 +← 1
        φ(q +← 1) ← (μ +← 1)
        3-RECTANGULATION
        q −← 1
        μ −← 1
        if not nonalignment desired then f_{x_q} > 0 −← 1, g_{y_q} > 0 −← 1, h_{z_q} > 0 −← 1
    ¶ colour q+1 with the 'same' colour as one of its neighbours if possible ¶
    if lmn − q > p − μ
    then
      if
        if nonalignment desired
        then true
        else
          if celltype(q+1) ≤ 6
          then true
          else
            ¶ test if the current partial 3-rectangulation is standard ¶
            [if (x_{q+1} = l−1) ∧ (y_{q+1} = m−1) then h_{z_q} > 0 else true] and
            [if (y_{q+1} = m−1) ∧ (z_{q+1} = n−1) then f_{x_q} > 0 else true] and
            [if (z_{q+1} = n−1) ∧ (x_{q+1} = l−1) then g_{y_q} > 0 else true]
      then
        ¶ S_{q+1} houses all the colour rules ¶
        S_{q+1} ← ∅
        if celltype(q+1) ≤ 3
        then
          S_{q+1} ← S_{q+1} ∪ case celltype(q+1) in
                  (= 1) {φ(q)}
                  (= 2) {φ(q+1−l)}
                  (= 3) {φ(q+1−lm)}
        else
          if celltype(q+1) ≤ 6
          then
            ⟨r, s, t⟩ ← case celltype(q+1) in
                  (= 4) ⟨φ(q), φ(q−l), φ(q+1−l)⟩
                  (= 5) ⟨φ(q), φ(q−lm), φ(q+1−lm)⟩
                  (= 6) ⟨φ(q+1−l), φ(q+1−lm−l), φ(q+1−lm)⟩
            if (r = t) ∨ (r ≠ s) then S_{q+1} ← S_{q+1} ∪ {r}
            if (r ≠ t) ∧ (s ≠ t) then S_{q+1} ← S_{q+1} ∪ {t}
          else
            ⟨r, s, t, r', t', q'⟩ ← ⟨φ(q), φ(q−l), φ(q+1−l), φ(q+1−lm), φ(q−lm),
                                                                   φ(q+1−lm−l)⟩
            if (r = t) ∨ (r = q') ∨ [(r ≠ s) ∧ (r ≠ r') ∧ (t ≠ q')] then S_{q+1} ← S_{q+1} ∪ {r}
            if (r ≠ t) ∧ (r ≠ q') ∧ {(t = q') ∨ [(s ≠ t) ∧ (t ≠ t')]} then S_{q+1} ← S_{q+1} ∪ {t}
            if (r ≠ t) ∧ (r ≠ q') ∧ (t ≠ q') ∧ (q' ≠ r') ∧ (q' ≠ t') then S_{q+1} ← S_{q+1} ∪ {q'}
        q +← 1
        for c ∈ S_q do
          φ(q) ← c
          3-RECTANGULATION
        q +← 1
```

Algorithm 3.

¶ backtrack ¶
if nonalignment desired

$$\textbf{then}\begin{cases} \P\ B_q,\ V_q, \text{ and } U_q \text{ are defined in TAGRULE } \P \\ \textbf{for } (j, w) \in B_q \textbf{ do } \begin{cases} j_{w_q} \ -\!\leftarrow 1 \\ j\mu_{w_q} \ -\!\leftarrow 1 \\ j\beta_q \leftarrow j\alpha_{w_q}(j\beta_q) \leftarrow 0 \end{cases} \\ \textbf{for } \langle \beta_1, \beta_2, (j, w) \rangle \in V_q \textbf{ do } j\alpha_{w_q}(\beta_1) \leftarrow \beta_2 \\ \textbf{for } (j, w) \in U_q \textbf{ do } j_{w_q} +\!\leftarrow 1 \end{cases}$$

return
¶ invoking main routine ¶
equivalence class ← 0
$\phi(q \leftarrow 1) \leftarrow \mu \leftarrow 1$
3-RECTANGULATION
¶ *equivalence class* contains the number of 3-rectangulations ¶

Algorithm 3 (continued).

introduced in the grid plane $W = w_q$. U_q also contains elements of the form (j, w), but in this case these elements signify that at level q two maximal planes in $W = w_q$ have been combined to form a single plane and consequently j_{w_q} has been decremented by unity. V_q contains elements of the form $\langle \beta_1, \beta_2, (j, w) \rangle$, which denote the fact that at level q the tag function has been modified; that is, $j\alpha_{w_q}(\beta_1)$ is mapped to β_2 before the tagging rule is applied. At each level q the sets B_q, U_q, and V_q are initially set equal to $\emptyset$. The procedure is Boolean-valued in that it returns a value of **true** if the grid planes are nonaligned—that is, contain exactly one maximal plane. (This condition is tested for only when q is the last face of a grid plane.)

Algorithm 3 gives the colouring grammars as a recursive backtrack procedure. There are two colouring steps in each stage q of the recursion. The first step describes the rules and conditions for assigning a 'new' colour to cell q. The second step describes the rules and conditions for assigning the 'same' colours as one of its neighbours to cell q. The rules are stored in a stack denoted by S_q. The procedure is a unification of the colouring grammars in that all possible combinations of trivalent, nonaligned, and general proper standard $[p, 3]$-rectangulations, for p fixed, on the given (l, m, n) unit grid may be generated.

The enumeration algorithm presented here is efficient in the following sense:
(1) each 3-rectangulation is uniquely constructed in a time step of $O[f(p)]$, where $f(p)$ is a polynomial in p;
(2) 'isomorph rejection'—that is, determining whether the rectangulation is canonical—is time-bounded by $O[g(p)]$, where $g(p)$ is a polynomial in p;
(3) the algorithm has a storage bound of $O[h(p)]$, where $h(p)$ is a polynomial in p.

The algorithm was implemented in ALGOL68C and run on the Cambridge Computer Laboratory's IBM370/165. The results of the enumeration for values of p up to 8 are given in the appendix. The program also enumerates the improper 3-rectangulations and thus provides further corroboration of the results in Bloch and Krishnamurti (1978). Earl's (1978) proposition that an $[l+m+n-2, 3]$-rectangulation is standard if and only if it is trivalent, nonaligned, and its graph contains no subgraphs isomorphic to the graph in figure 2 was computationally verified for values of p up to 8.

d-rectangulations

The colouring method developed in the preceding sections may be extended to enumerate assemblies of d-dimensional boxes into a d-dimensional box, $d \geqslant 4$.

A *d-rectangulation* is an arrangement of an $(l_1, l_2, ..., l_d)$ unit grid into nonoverlapping d-rectangles whose $(d-1)$-faces lie on the grid $(d-1)$-planes. These faces are combined to form *maximal* $(d-1)$*-planes*, which are higher-dimensional analogues of maximal planes. A d-rectangulation is *proper* if $l_i \geqslant 2$ for all i; *standard* if each grid hyperplane contains at least one maximal $(d-1)$-plane; and *nonaligned* if each grid hyperplane contains at most one maximal $(d-1)$-plane. Let a cross section be a hyperplanar segment parallel to one of the grid hyperplanes and which does not contain any maximal $(d-1)$-planes. A d-rectangulation is *trivalent* if each of its cross sections is a trivalent $(d-1)$-rectangulation. A $[p, d]$*-rectangulation* is a d-rectangulation with *content* p; that is, the number of component d-rectangulations is p.

Two d-rectangulations are *equivalent* if they are identical after a sequence of 'right-handed' rotations or reflections. From symmetry considerations it may be assumed that $l_1 \geqslant l_2 \geqslant \ldots \geqslant l_d$. Consequently two d-rectangulations are equivalent if they may be transformed into one another by a transformation that leaves the unit grid invariant in space. The group of rotations that leave the unit grid invariant in space is isomorphic to an even subgroup of O_d, the orthogonal group of rotations of a d-cube. Littlewood (1931) describes the generators of O_d, and Krishnamurti and Roe (1979) present a method for representing the elements of O_d as coordinate-coordinate mappings.

The grid cells may be associated with integer coordinates $(x^1, x^2, \ldots, x^d) \in \mathop{\times}\limits_i Z^+_{l_i}$ Each grid cell $\chi_q = (x^1_q, x^2_q, \ldots, x^d_q)$ may be uniquely labelled—called its Γ-number—by the mapping

$$\Gamma(\chi_q) = 1 + \sum_{i \geqslant 1}\left(\prod_{0 \leqslant j < i} l_j\right)x^i_q = q \;,$$

where l_0 is assumed to be 1. Each χ_q may be assigned a coordinate *word* $\Psi_q = \psi^1_q\psi^2_q \ldots \psi^d_q$ radix 2, where $\psi^i_q = 1$ if $x^i_q > 0$ and $\psi^i_q = 0$ otherwise. $\sum_i \psi^i_q$ is referred to as the *weight*, ω_q, of the word Ψ_q. There are $\binom{d}{k}$ distinct words of weight k, each corresponding to a distinct *cell type*. Let $A(q) = \left\{\prod_{0 \leqslant j < i} l_i \colon\; \psi^i_q = 1\right\}$ denote the adjacency set of cell q. $A(q)$ may be described by the ordered set $\Xi_q = \langle \xi_1, \xi_2, \ldots, \xi_{\omega_q} \rangle$, where $\xi_i < \xi_j$ whenever $i < j$, and $\xi_i \in A(q)$ for all i. Ξ_q has the following interpretation: if $\xi \in \Xi_q$ then $q-\xi$ is the Γ-number of an adjacent neighbour of q. Ξ_q contains the list of all adjacent neighbours of q with lower Γ-numbers.

A d-rectangulation R may be represented by its ϕ-colouring,

$$\phi(R) \equiv \left\langle \phi(1), \phi(2), \ldots, \phi\left(\prod_i l_i\right)\right\rangle \;,$$

defined in the usual manner. R is *canonical*—that is, the representative of its equivalence class under T, the group that leaves the unit grid invariant in space—if and only if $\phi(R) \leqslant_{\mathrm{L}} \phi(\tau R)$ for all $\tau \in T$.

The basic enumeration problem is to determine all proper standard nonequivalent $[p, d]$-rectangulations, for p fixed. The enumeration may be described by the following recursive procedure. Construct all possible colourings $\langle \phi(1), \phi(2), \ldots, \phi(q) \rangle$ from the current colouring $\langle \phi(1), \phi(2), \ldots, \phi(q-1) \rangle$ until $q = \prod_i l_i$. Each stage of the recursion colours cell q. The colour rules may be of two kinds: rules that colour q differently from the previous $q-1$ cells and rules that colour q the same as one or more of the $q-1$ cells. Clearly in the latter case q must be coloured the same as one of its neighbouring adjacent cells.

Consider any stage of the recursion. Let it be q. Furthermore let the cells be coloured in order of their increasing Γ-numbers. Let ω_q be the weight of Ψ_q. Then there are $\omega_q + 1$ possible colouring rules that apply. In general the colouring grammar for d-rectangulations may be described by $\sum_{i=0} (i+1) = \binom{d+2}{2}$ colouring rules, with $k+1$ rules for cells with weight k. Let Ξ_q represent the adjacency set for q.

Rule 1: q is coloured differently from the previous q − 1 cells
Let μ be the maximum number of colours used so far. Clearly q may be assigned a new colour provided $p > \mu$. Moreover q may be assigned a new colour provided the neighbourhood of colours around q does not force q to be coloured the same as one of its adjacent cells. Consider the set of predicates $\{F_i\}$, each of which is defined as follows:

$$F_i = \bigcap_{j > i} [\phi(q-\xi_i) \neq \phi(q-\xi_j)] \,, \quad \text{for all } i < \omega_q \,.$$

Also define

$$F = \bigcap_{i < \omega_q} F_i \,.$$

Then q is given a new colour provided F holds: $\phi(q) \leftarrow (\mu + \leftarrow 1)$ if F is **true**.

Suppose trivalent d-rectangulations are desired. In this case additional predicates have to be described. Let

$$Q_{ij} = [\phi(q-\xi_i) = \phi(q-\xi_i-\xi_j)] \,, \qquad P_{ij} = Q_{ij} \cup Q_{ji} \,,$$

$$P_i = \bigcap_{j > i} P_{ij} \,, \qquad P = \bigcap_{i < \omega_q} P_i \,.$$

The satisfaction of P ensures that in each plane containing q, $q-\xi_i$, $q-\xi_j$, and $q-\xi_i-\xi_j$, ξ_i, $\xi_j \in \Xi_q$, four colours do not meet at a point, which would violate the trivalency requirement. Hence for trivalent d-rectangulations q may be coloured by a new colour provided P holds (in addition to F).

Rule 2: q is coloured the same as one of its adjacent neighbours
In this case q has a choice of ω_q colours corresponding to the ω_q neighbours in Ξ_q. There are two cases to consider.
Case 2.1: F does not hold. That is, there are at least two neighbours $q-\xi_i$ and $q-\xi_j$ such that $\phi(q-\xi_i) = \phi(q-\xi_j)$. In this case q must be coloured the same as these two cells.
Case 2.2: F holds. In this case q may be coloured the same as $\phi(q-\xi_i)$ provided $q-\xi_i$ does not have the same colour as one of its neighbours in the plane containing q. Let $G_i = \bigcap_{j \neq i} \bar{Q}_{ij}$, where $\bar{Q}$ denotes the complement of Q. The colouring rule states: q may be assigned the colour $\phi(q-\xi_i)$ provided G_i holds.
Both cases may be considered only if there is a sufficient number of grid cells remaining to ensure that eventually p colours are used. That is, rule 2 applies provided $\prod_i l_i - q \geqslant p - \mu$.

For the d-rectangulations to be standard the last cell q on any grid hyperplane must be coloured in such a way that there is at least one $(d-1)$-face in that plane. Let $\{f_{i,x_q^i}\}$ denote the set of grid variables which indicate the absence or presence of a maximal $(d-1)$-plane in the grid plane $X_j = x_q^i$. Every application of rule 1 introduces a face in the grid planes $X_j = x_q^j$ provided $\psi_q^j = 1$ in Ψ_q. If $\chi_q = (l_1-1, l_2-1, ..., l_{j-1}-1, x_q^j, l_{j+1}-1, ..., l_d-1)$, then before the application of rule 2 it must be ensured that $f_{j,x_q^i} > 0$.

Algorithm 4 provides in ALGOL-like notation a translation of these rules for generating general and trivalent proper standard $[p, d]$-rectangulations, for p fixed, on a given $(l_1, l_2, ..., l_d)$ unit grid. Nonaligned d-rectangulation may be obtained by defining local tagging rules for the grid faces in a manner similar to that done for 3-rectangulations.

Algorithm *d*-RECTANGULATION

```
¶ Routine to generate general and trivalent standard [p, d]-rectangulations, for p fixed, on the given
  (l_1, l_2, ..., l_d) unit grid ¶
if q = ∏_i l_i
then  if CANONICAL (generated d-rectangulation) then equivalence class +← 1
else
  q +← 1
  ¶ Compute the predicates F, P, and {G_i; 1 ⩽ i ⩽ ω_q}, which are initially set to true;
    c* corresponds to the case when F does not hold, and consequently there are at least two
    neighbours q − ξ_i and q − ξ_j such that the colours of both are equal, in which case q is
    forced to take on the colour of c* ¶
  for i ∈ ⟨1, 2, ..., ω_q − 1⟩ do
      r_i ← φ(q − ξ_i)
      for j ∈ ⟨i+1, ..., ω_q⟩ do
          r_j ← φ(q − ξ_j)
          r_ij ← φ(q − ξ_i − ξ_j)
          if P then P ← (r_i = r_ij) ∨ (r_ij = r_j)
          if F then if not (F ← r_i ≠ r_j) then c* ← r_i
          if G_i then G_i ← r_i ≠ r_ij
          if G_j then G_j ← r_j ≠ r_ij
  ¶ q is assigned a new colour if possible ¶
  if p > μ and F and (if trivalency required then P else true)
  then
      for i ∈ ⟨1, 2, ..., d⟩ do if ψ^i_q = 1 then f_{i, x^i_q} +← 1
      φ(q) ← (μ +← 1)
      d-RECTANGULATION
      μ −← 1
      for i ∈ ⟨1, 2, ..., d⟩ do if ψ^i_q = 1 then f_{i, x^i_q} −← 1
  ¶ q is assigned the same colour as one of its neighbours if possible ¶
  if ∏_i l_i − q ⩾ p − μ
  then
      if
          ¶ test if the d-rectangulation is standard ¶
          if ω_q ≠ d
          then true
          else
              if ∏_i l_i = q
              then ⋂_i (f_{i, x^i_q} > 0)
              else
                  if there exists a j such that x^j_q < l_j − 1 and, for all i ≠ j, x^i_q = l_i − 1
                  then f_{j, x^j_q} > 0
                  else true
      then
          ¶ stack contains the colour rules; and is initially empty;
            ⇐ denotes a 'push' into stack;
            ⇒ denotes a 'pop' from stack ¶
          if not F
          then stack ⇐ c*
          else for i ∈ ⟨1, 2, ..., ω_q⟩ do if G_i then stack ⇐ φ(q − ξ_i)
          while stack is not empty do
              φ(q) ← [stack ⇒ φ(q)]
              d-RECTANGULATION
  q −← 1
return
¶ invoking routine ¶
equivalence class ← 0
φ(q ← 1) ← μ ← 1
d-RECTANGULATION
```

Algorithm 4.

Acknowledgements. This work has been partially supported by The Open University Research Fund and the Science Research Council.

References

Bloch C J, Krishnamurti R, 1978 "The counting of rectangular dissections" *Environment and Planning B* **5** 207-214

Earl C F, 1978 "Joints in two- and three-dimensional rectangular dissections" *Environment and Planning B* **5** 179-187

Earl C F, March L J, 1979 "Architectural applications of graph theory" in *Applications of Graph Theory* Eds R J Wilson, L W Beineke (Academic Press, London) pp 327-355

Flemming U, 1978 "Wall representations of rectangular dissections and their use in automated space allocation" *Environment and Planning B* **5** 215-232

Krishnamurti R, Roe P H O'N, 1978 "Algorithmic aspects of plan generation and enumeration" *Environment and Planning B* **5** 157-177

Krishnamurti R, Roe P H O'N, 1979 "On the generation and enumeration of tessellation designs" *Environment and Planning B* **6** 191-260

Littlewood D E, 1931 "The groups of the regular solids in *n*-dimensions" *Proceedings of the London Mathematical Society* **32** 10-20

March L J, Earl C F, 1977 "On counting architectural plans" *Environment and Planning B* **4** 57-80

Tarjan R E, 1972 "Depth first search and linear graph algorithms" *SIAM Journal on Computing* **1** 146-160

APPENDIX

Table A1. The breakdown of standard 3-rectangulations, for contents up to and including $p = 8$, according to their unit grids.

p	Unit grid	Number of 3-rectangulations			
		general	nonaligned	trivalent	fundamental
5	improper	24	23	22	21
	(2,2,2)	5	5	2	2
	(3,2,2)	22	22	22	22
	all	51	50	46	45
6	improper	126	119	108	101
	(2,2,2)	4	4	0	0
	(3,2,2)	70	38	27	21
	(4,2,2)	79	79	79	79
	(3,3,2)	159	159	159	159
	all	438	399	373	360
7	improper	815	735	668	591
	(2,2,2)	1	1	0	0
	(3,2,2)	118	72	12	8
	(4,2,2)	424	376	177	129
	(5,2,2)	276	276	276	276
	(3,3,2)	931	810	411	290
	(4,3,2)	1844	1844	1844	1844
	(3,3,3)	548	548	548	548
	all	4957	4662	3936	3686
8	improper	6465	5527	5026	4168
	(2,2,2)	1	1	0	0
	(3,2,2)	123	114	0	0
	(4,2,2)	1194	1032	148	84
	(5,2,2)	2211	1907	924	620
	(6,2,2)	900	900	900	900
	(3,3,2)	3102	2580	449	244
	(4,3,2)	17066	14295	7934	5163
	(5,3,2)	9740	9740	9740	9740
	(4,4,2)	8241	8241	8241	8241
	(3,3,3)	5709	4780	2621	1692
	(4,3,3)	13680	13680	13680	13680
	all	68432	62797	49663	44532