Environment and Planning B, 1981, volume 8, pages 5-40

The construction of shapes

R Krishnamurti
Centre for Configurational Studies, The Open University, Milton Keynes MK7 6AA, England
Received 28 November 1980, in revised form 24 February 1981

Abstract. An algorithm for shape rule application is presented.

A shape rule $\alpha \rightarrow \beta$ applies to a labelled shape γ whenever there is a transformation τ that makes α a subshape of γ. In this case, a new labelled shape can be obtained by replacing the occurrence of $\tau(\alpha)$ in γ with $\tau(\beta)$. The algorithm required for this process is developed in this paper. This algorithm determines all possible distinct transformations under which a given shape rule applies to a given labelled shape and the corresponding labelled shapes resulting from such applications. The definitions and notations given for labelled shapes and shape grammars by Stiny (1980) are used.

Euclidean transformations

The transformations of *translation*, *rotation*, *reflection*, *scale*, and finite *compositions* of these are referred to as the euclidean transformations, which are hereinafter referred to simply as transformations, denoted by τ. A transformation τ can be expressed, in two dimensions, as a mapping τ: $\mathrm{R}^2 \rightarrow \mathrm{R}^2$, where R is the set of reals. Furthermore, τ can be composed as the ordered pair of mappings, $\tau = \langle \tau_x, \tau_y \rangle$, where τ_x and τ_y each take the form τ_z: $\mathrm{R}^2 \rightarrow \mathrm{R}$, $z \in \{x, y\}$. That is, τ is described by the mapping τ: $\langle x, y \rangle \rightarrow \langle \tau_x(x, y), \tau_y(x, y) \rangle$, where $\langle x, y \rangle$ represent the coordinates of a point. The expression $\langle \tau_x(x, y), \tau_y(x, y) \rangle$ represents the coordinates of the transformed point. The transformation τ is *linear* if and only if both τ_x and τ_y can be expressed as polynomials over the reals, having the form: $ax+by+c$. The coefficients a, b, and c are constants dependent on τ.

The general expressions for the plane transformations are listed below:

(1) Translation [through t_x units in the X-direction and t_y units in the Y-direction]:
$\langle x, y \rangle \rightarrow \langle x+t_x, y+t_y \rangle$.

(2) Rotation [about the origin through a counterclockwise angle of θ]:
$\langle x, y \rangle \rightarrow \langle ax-by, ay+bx \rangle$, where $a = \cos\theta$ and $b = \sin\theta$.

(3) Reflection [about (a) the X-axis, and (b) the Y-axis]:
(a) $\langle x, y \rangle \rightarrow \langle x, -y \rangle$,
(b) $\langle x, y \rangle \rightarrow \langle -x, y \rangle$.

(4) Scale [or change of size through a scale factor, $c > 0$]:
$\langle x, y \rangle \rightarrow \langle cx, cy \rangle$.

A transformation consisting of a finite sequence of transformations is a *composition*. The sequence $(\tau_1, ..., \tau_j, ..., \tau_n)$, where each τ_j, $1 \leqslant j \leqslant n$, is one of the above listed transformations, denotes the composed transformation $\tau_1(\tau_2(...(\tau_j(...(\tau_n(\ ))...))...))$. Therefore, in general, a transformation τ can be expressed as a mapping of the form τ: $\langle x, y \rangle \rightarrow \langle a_x x + b_x y + c_x, a_y x + b_y y + c_y \rangle$, which is clearly linear.

The transformation τ of a labelled shape σ, $\sigma = \langle s, P \rangle$, is the labelled shape denoted by $\tau(\sigma)$, $\tau(\sigma) = \langle \tau(s), \tau(P) \rangle$, which is obtained by changing the spatial disposition and/or size of σ. More precisely, $\tau(\sigma)$ is defined as follows. Let p,

$p = \langle x, y \rangle$, denote a point. Then:

$$\tau(p) = \langle \tau_x(x, y), \tau_y(x, y) \rangle ,$$

$$\tau(s) = \{\{\tau(p_1), \tau(p_2)\} \text{ is a maximal line in } \tau(s) | \{p_1, p_2\} \text{ is a maximal line in } s\} ,$$

$$\tau(P) = \{\tau(p){:}A \text{ is a labelled point in } \tau(P) | p{:}A \text{ is a labelled point in } P\} .$$

In other words, τ takes each point, maximal line, and labelled point in σ to a corresponding point, maximal line, and labelled point in $\tau(\sigma)$; τ is bijective in the sense that there is a transformation τ^{-1} which satisfies $\sigma = \tau[\tau^{-1}(\sigma)] = \tau^{-1}[\tau(\sigma)]$. It should be noted that τ does not alter the labels associated with the labelled points. Figure 1 presents a labelled shape σ and examples of possible transformations $\tau(\sigma)$.

Recall that the line descriptor (Krishnamurti, 1980) of a line is given by the pair $\langle \mu, \nu \rangle$, where μ is the slope of line l and ν is the y-intercept for l in the case that l is not vertical and the x-intercept for l otherwise. The transformation $\tau(l)$ of l has the line descriptor $\langle \tau(\mu), \tau(\nu) \rangle$, which is obtained in the following way. Let τ be the mapping $\langle a_x x + b_x y + c_x, a_y x + b_y y + c_y \rangle$, then:

Case 1: $\mu \neq \infty$ (the lines are nonvertical)

$$\left.\begin{aligned} \tau(\mu) &= \frac{\mu a_x - a_y}{b_y - \mu b_x} , \\ \tau(\nu) &= \begin{cases} \dfrac{\mu c_x - c_y + \nu}{(b_y - \mu b_x)} , & \text{for } b_y - \mu b_x \neq 0 , \\ \dfrac{\mu c_x - c_y + \nu}{(a_y - \mu a_x)} , & \text{otherwise.} \end{cases} \end{aligned}\right\} \tag{1}$$

Case 2: $\mu = \infty$ (the lines are vertical)

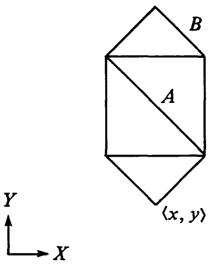
$$\left.\begin{aligned} \tau(\mu) &= \frac{a_x}{b_x} , \\ \tau(\nu) &= \begin{cases} \dfrac{\nu - c_x}{b_x} , & \text{for } b_x \neq 0 , \\ \dfrac{\nu - c_x}{a_x} , & \text{otherwise.} \end{cases} \end{aligned}\right\} \tag{2}$$

Finally, transformation τ preserves either the order or the antiorder of the lines in a list of multiple colinear lines. As shown in Krishnamurti (1980), any shape s can be represented by a shape union: $s = s_1 + ... + s_m$, where each shape s_k, $1 \leqslant k \leqslant m$, consists of multiple colinear maximal lines. Moreover, each such s_k can be represented as an ordered list of lines: $L_k = \langle l_1, ..., l_j, ..., l_n \rangle$, where $l_1 < ... < l_j < ... < l_n$. The relation $<$ on maximal lines is defined as follows. Let $p = \langle x, y \rangle$ denote a general point. Then, for distinct points p_1 and p_2, $p_1 < p_2$ if and only if either $x_1 < x_2$ or $x_1 = x_2$ and $y_1 < y_2$. Each line l is represented by an ordered pair of distinct end points denoted by $l = \langle \text{tail}, \text{head} \rangle$ where tail of $l <$ head of l. Then, for any two consecutive members l_j, l_{j+1}, $1 \leqslant j < n$, in L_k, for each s_k, head of $l_j <$ tail of l_{j+1}. (This is always the case since colinear maximal lines in same shape do not overlap.) Therefore, the corresponding list of multiple colinear lines $\tau(L_k) = \langle \tau(l_1), ..., \tau(l_j), ..., \tau(l_n) \rangle$, $1 \leqslant j \leqslant n$, must satisfy either

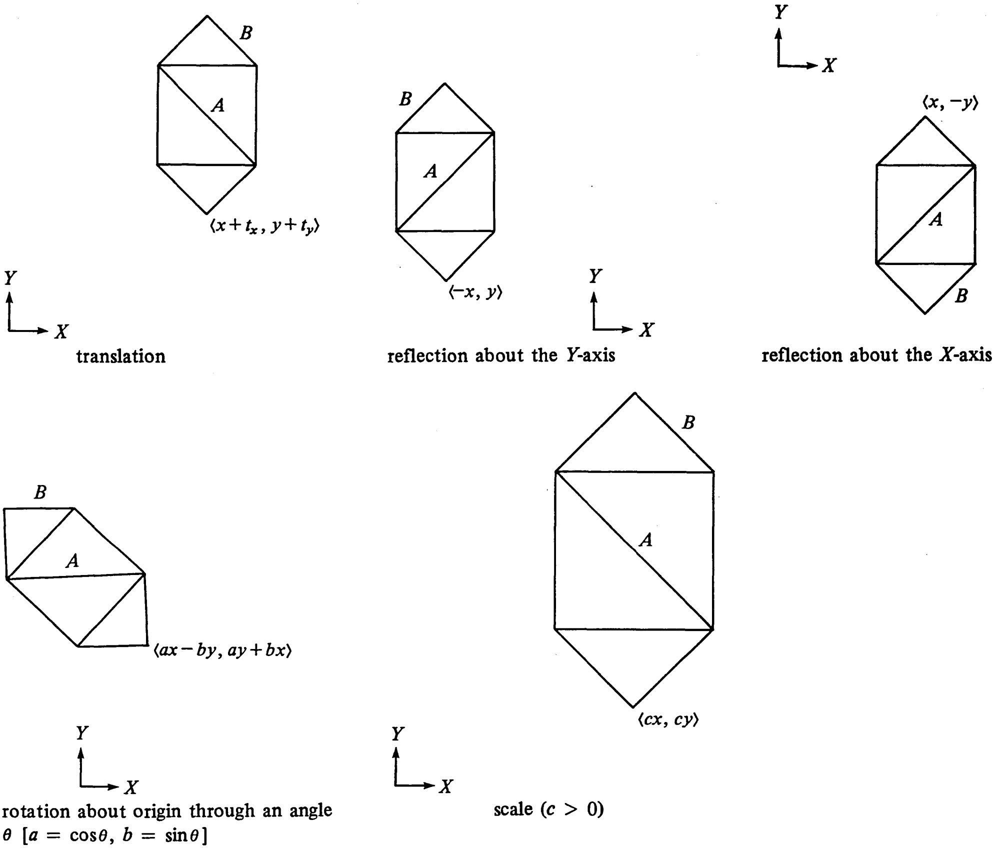
$$\text{tail of } \tau(l_j) = \tau(\text{tail of } l_j) , \qquad \text{head of } \tau(l_j) = \tau(\text{head of } l_j) , \tag{3}$$

in which case $\tau(l_1) < ... < \tau(l_j) < ... < \tau(l_n)$; or

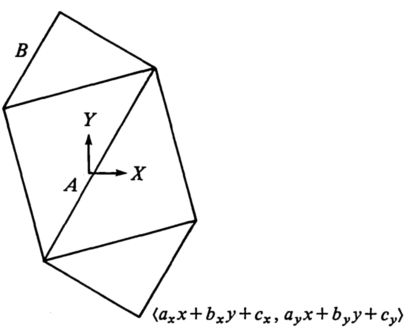
$$\text{tail of } \tau(l_j) = \tau(\text{head of } l_j) , \qquad \text{head of } \tau(l_j) = \tau(\text{tail of } l_j) , \tag{4}$$



(a) A labelled shape



(b) Examples of euclidean transformations on the labelled shape



(c) A composition of euclidean transformations

Figure 1.

in which case $\tau(l_n) < ... < \tau(l_j) < ... < \tau(l_1)$. Thus, the algorithms described in Krishnamurti (1980) for subshape ($\leq$), shape identity ($=$), shape union ($+$), difference ($-$), and intersection ($\cdot$) can be applied to transformations of labelled shapes without additional computational effort. These algorithms are used in the shape rule application algorithm presented below.

Shape rules

Shape rules provide the basis for the recursive construction of shapes. A shape rule takes the form $\alpha \rightarrow \beta$, where α and β are labelled shapes. A shape rule is initially represented by the ordered pair R, $R = \langle\alpha, \beta\rangle$. R applies to a labelled shape γ whenever there is a transformation τ such that $\tau(\alpha) \leq \gamma$. In other words, α is similar to some part of γ. A new labelled shape γ^* is obtained from γ by applying R under τ, when γ^* satisfies the expression $\gamma^* \leftarrow [\gamma - \tau(\alpha)] + \tau(\beta)$.

For computational completeness (and consistency) a shape rule is allowed to apply to any labelled shape γ. In this case, the previous expression for γ^* becomes

$$\gamma^* \leftarrow \begin{cases} [\gamma - \tau(\alpha)] + \tau(\beta) , & \text{if } R \text{ applies to } \gamma \text{ under } \tau, \\ \gamma , & \text{otherwise.} \end{cases}$$

The application of a shape rule R, $R = \langle\alpha, \beta\rangle$, to a labelled shape is outlined by the following general procedure.

Procedure

Step 1: Determine if R applies to γ for some transformation τ.

Step 2.1: If so, perform the shape operations: (a) take the shape difference of γ and $\tau(\alpha)$; (b) take the shape union of the labelled shape produced from (a) and $\tau(\beta)$; then γ^* is the labelled shape resulting from (b).

Step 2.2: Otherwise, the shape rule does not apply, and γ^* equals γ.

In practice, step 2.1 is computationally wasteful. The reason for this is that the shape intersection $\alpha \cdot \beta$ is generally nonempty. The maximal lines and labelled points in $\tau(\alpha \cdot \beta)$ are examined twice, once each for the shape difference and the shape union. In step 2.1(a) $\tau(\alpha \cdot \beta)$ is part of the shape removed from γ, and in step 2.1(b), $\tau(\alpha \cdot \beta)$ is part of the shape added to produce γ^*. This follows from the fact that since $\alpha \cdot \beta$ is a subshape both of α and of β, $\tau(\alpha \cdot \beta)$ is a subshape both of $\tau(\alpha)$ and of $\tau(\beta)$. In other words, $\tau(\alpha \cdot \beta)$ is a subshape both of γ and of γ^*. This duplication of computational effort can easily be eliminated as follows. Any labelled shape γ can be described by the shape union

$$\gamma = [\gamma - \delta] + \gamma \cdot \delta = \gamma \cdot \delta + [\gamma - \delta]$$

where δ is any other labelled shape. Therefore, the expression $[\gamma - \tau(\alpha)] + \tau(\beta)$ can be rewritten as $[\gamma - \tau(\alpha - \beta) - \tau(\alpha \cdot \beta)] + \tau(\alpha \cdot \beta) + \tau(\beta - \alpha)$, or as $[\gamma - \tau(\alpha - \beta)] + \tau(\beta - \alpha)$. Thus, the expression for shape rule application reduces to

$$\gamma^* \leftarrow \begin{cases} [\gamma - \tau(\alpha - \beta)] + \tau(\beta - \alpha) , & \text{if } R \text{ applies to } \gamma \text{ under } \tau, \\ \gamma , & \text{otherwise.} \end{cases}$$

Since the shape differences $\alpha - \beta$ and $\beta - \alpha$ have an empty shape intersection, this expression is optimal. Consequently, a shape rule R may be represented by the ordered triple $\langle\alpha, \alpha - \beta, \beta - \alpha\rangle$. Figure 2 presents examples of typical shape rules taken from Stiny and Mitchell (1980) and the extent of the computational saving that results by using the above expression.

The *backtracking* identity is now introduced. Suppose the shape rule R, $R = \langle\alpha, \alpha - \beta, \beta - \alpha\rangle$, applies to the current labelled shape γ. Let the shape

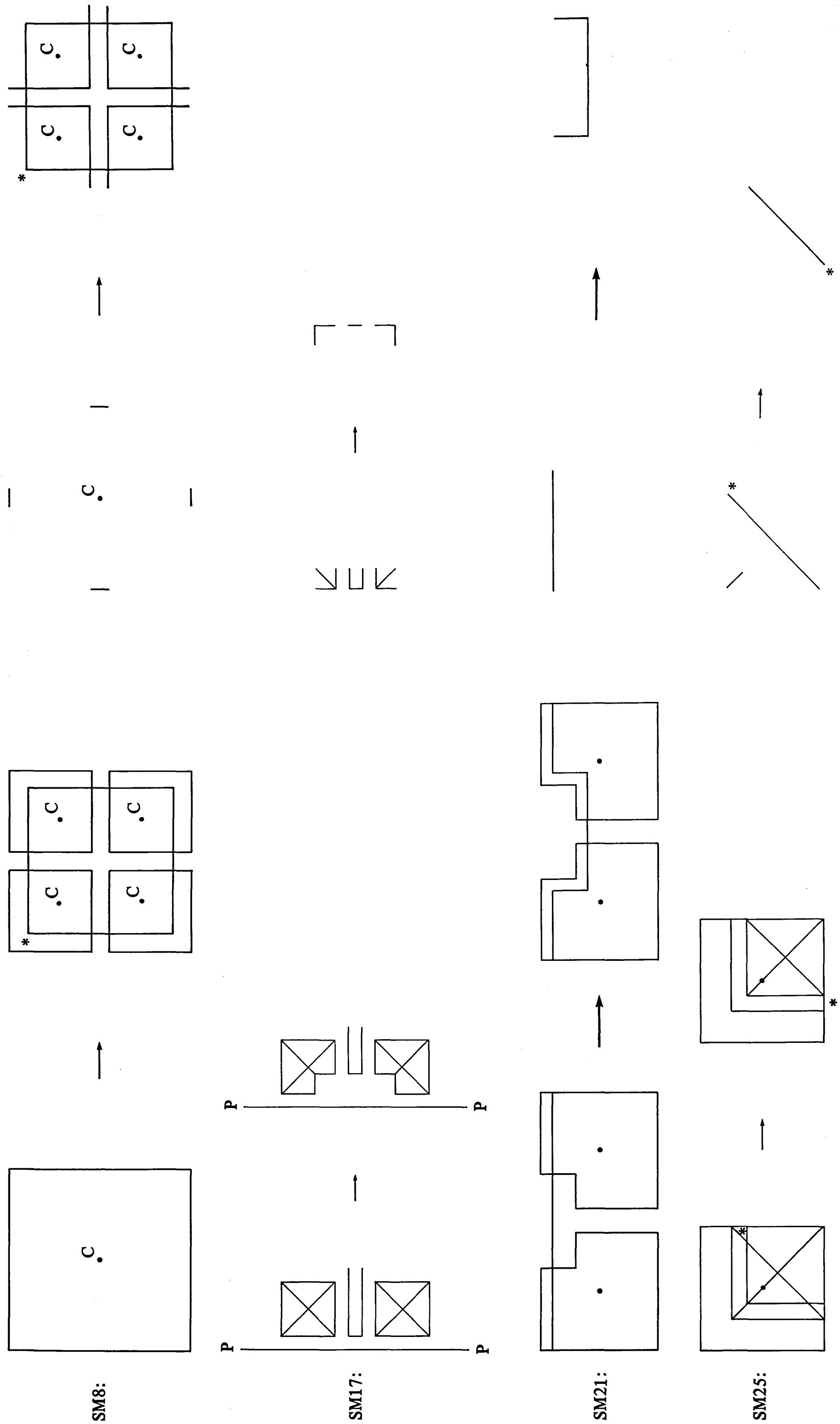


Figure 2. Some typical shape rules (not to scale). The numbers prefixed by SM refer to the corresponding shape rule numbers in Stiny and Mitchell (1980). The left-hand figure gives the shape rule $\alpha \rightarrow \beta$, and the right-hand figure the shape rule $\alpha - \beta \rightarrow \beta - \alpha$.

intersection β^* be given by the expression $\beta^* \leftarrow \gamma \cdot \tau(\alpha - \beta)$. Then, one has $\gamma = [\gamma^* - \tau(\beta - \alpha)] + \tau(\alpha - \beta) + \beta^*$. This identity is particularly useful for computer implementations of the shape grammar formalism. It often happens that designers discover that a particular sequence of shape rule applications yields an undesirable shape, in which case by 'backtracking' through this sequence they can find a shape from which to restart a fresh sequence of shape applications. Clearly, in order to incorporate the backtracking facility it is necessary to keep a history of the β^*s for each rule application. Fortunately, in practice, β^* is often the empty shape $\langle s_\phi, \emptyset \rangle$, and thus, implementing the backtracking facility requires little additional storage.

The subshape detection problem

The hardest and certainly the most crucial step in the application of a shape rule to a labelled shape is in actually determining whether or not the shape rule applies to the labelled shape. In general, there may be several subshapes in a given labelled shape γ to which a given shape rule may be applied. That is, γ may contain subshapes $\gamma_1, \gamma_2, ...,$ each of which is similar to α. Thus, there may be a list of distinct transformations, $\tau_1, \tau_2, ...,$ such that $\tau_j(\alpha) = \gamma_j \leq \gamma$. The set of all such transformations under which the shape rule R applies to the labelled shape γ is denoted by $T_{R,\gamma}$. When R does not apply to γ, $T_{R,\gamma} = \emptyset$.

In order to define $T_{R,\gamma}$ for any given shape rule R, $R = \langle \alpha, \alpha - \beta, \beta - \alpha \rangle$, and any given labelled shape γ, the following two problems are considered:

1. Suppose we are given a transformation τ under which the labelled shape α is similar to a subshape of the labelled shape γ. Can we find a computation for τ? That is, we are given the correspondence between the points and lines in α and an equal number of points and lines in γ. We are required to obtain the coefficients of the transformation that represents the given correspondence.
2. Given that we have a method for computing τ, can we generate all valid transformations τ that satisfy $\tau(\alpha) \leq \gamma$? We may restate this problem as follows. Suppose the method for determining the coefficients of τ relies on some relationship. Can we combinatorially derive all valid instances of this relationship?

Problem 1: the determination of τ

Consider a transformation τ, $\tau = \langle \tau_x, \tau_y \rangle$, which satisfies $\tau(\alpha) \leq \gamma$. Since τ is linear, we may let $\tau_x = a_x x + b_x y + c_x$, and $\tau_y = a_y x + b_y y + c_y$, where the as, bs, and cs are as yet undetermined. Suppose p, $p = \langle x, y \rangle$, denotes a general point associated with α. For example, for some label A, $p{:}A$ is a labelled point in α. Or, then again, p may be an end point of a maximal line in α, or p may be some distinguishable point coincident with a maximal line in α. Then, τ maps each such point p associated with α to a corresponding point $\tau(p)$, where $\tau(p) = p' = \langle x', y' \rangle$, associated with γ. Then,

$$a_x x + b_x y + c_x = x' , \tag{5}$$

$$a_y x + b_y y + c_y = y' . \tag{6}$$

Equations (5) and (6) are each in three unknowns, and so the coefficients can be uniquely and completely solved provided we have three distinct points p_1, p_2, p_3 associated with α which correspond to three distinct points p'_1, p'_2, p'_3 associated with γ.

The solutions of the matrix equations:

$$\begin{bmatrix} x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \\ x_3 & y_3 & 1 \end{bmatrix} \begin{bmatrix} a_x \\ b_x \\ c_x \end{bmatrix} = \begin{bmatrix} x'_1 \\ x'_2 \\ x'_3 \end{bmatrix} , \tag{7}$$

and

$$\begin{bmatrix} x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \\ x_3 & y_3 & 1 \end{bmatrix} \begin{bmatrix} a_y \\ b_y \\ c_y \end{bmatrix} = \begin{bmatrix} y'_1 \\ y'_2 \\ y'_3 \end{bmatrix} , \qquad (8)$$

determine τ uniquely provided the 3×3 matrix is nonsingular; that is, the points p_1, p_2, and p_3 must not all lie in a straight line. Thus no two lines in the set of lines $\{\{p_1, p_2\}, \{p_2, p_3\}, \{p_3, p_1\}\}$ are colinear, and the lines in this set form a triangle. The set of lines $\{\{p'_1, p'_2\}, \{p'_2, p'_3\}, \{p'_3, p'_1\}\}$ must also form a triangle. Moreover, these triangles must be similar.

Now the algorithms already developed in Krishnamurti (1980) as well as the ones presented later in this paper are given for rational shapes only. Thus, the coefficients derived for a transformation τ such that $\tau(\alpha) \leq \gamma$ will also be rational. For convenience, these coefficients are maintained in their primitive form. That is, for a rational r given by $\langle r_n, r_d \rangle$, the integers r_n and r_d are relatively prime. The expressions (5) and (6) thus take form

$$\langle a_n, a_d \rangle \langle x_{jn}, x_{jd} \rangle + \langle b_n, b_d \rangle \langle y_{jn}, y_{jd} \rangle + \langle c_n, c_d \rangle = \langle z'_{jn}, z'_{jd} \rangle , \qquad (9)$$

where a, b, c, z, respectively, denote either a_x, b_x, c_x, and x, or a_y, b_y, c_y, and y. Expression (9) can be expanded and rewritten as

$$\langle a_n, a_d \rangle \underbrace{x_{jn} y_{jd} z'_{jd}}_{A_j} + \langle b_n, b_d \rangle \underbrace{x_{jd} y_{jn} z'_{jd}}_{B_j} + \langle c_n, c_d \rangle \underbrace{x_{jd} y_{jd} z'_{jd}}_{C_j} = \underbrace{z'_{jn} x_{jd} y_{jd}}_{D_j} . \qquad (10)$$

For $j = 2, 3$, define the quantities

$$¤_j(B, A) = B_1 A_j - B_j A_1 , \quad ¤_j(C, A) = C_1 A_j - C_j A_1 , \quad ¤_j(D, A) = D_1 A_j - D_j A_1 . \qquad (11)$$

Then,

$$\left. \begin{aligned} \langle c_n, c_d \rangle &\leftarrow_p \langle ¤_2(D,A)¤_3(B,A) - ¤_3(D,A)¤_2(B,A),\ ¤_3(B,A)¤_2(C,A) - ¤_2(B,A)¤_3(C,A) \rangle \\ \langle b_n, b_d \rangle &\leftarrow_p \langle ¤_2(D,A)¤_3(C,A) - ¤_3(D,A)¤_2(C,A), \\ &\qquad -[¤_3(B,A)¤_2(C,A) - ¤_2(B,A)¤_3(C,A)] \rangle \\ \langle a_n, a_d \rangle &\leftarrow_p \langle D_1 b_n c_d - C_1 b_d c_n - B_1 b_n c_d ,\ A_1 b_d c_d \rangle , \end{aligned} \right\} \qquad (12)$$

where $\leftarrow_p$ indicates that for each of expressions (12) the left-hand side is assigned the primitive form of the right-hand side of the expression. [The reader is referred to the section on rational shapes, in Krishnamurti (1980)(1).]

The above procedure involves purely integer computation: hence, the values for the coefficients of τ_x and τ_y are exact in the sense that they do not have to be stored internally in a computer as finite approximations of real numbers.

A correspondence between three distinct points of α and three distinct points of γ yields a unique transformation τ for which the relation $\tau(\alpha) \leq \gamma$ may hold. Two such transformations are possible whenever a correspondence between pairs of specified distinct points associated with α and γ is used.

Suppose τ is a transformation such that $\tau(\alpha) \leq \gamma$. Since τ maps the labelled shape α to a similar labelled shape $\tau(\alpha)$, it must also map any point *relative* to α to a corresponding point which bears the *same* relationship to $\tau(\alpha)$. In other words, τ maps

(1) In that paper, the procedure for determining the primitive form of the rational $\langle r_n, r_d \rangle$ is given for $r_d > 0$. In fact, it should read $r_d \neq 0$.

any point relative to the line $\langle p_1, p_2 \rangle$ to a corresponding point which bears the same relationship to $\langle p'_1, p'_2 \rangle$, where $p'_1 = \tau(p_1)$ and $p'_2 = \tau(p_2)$. Therefore, all one has to do is to select two similar triangles, one relative to $\langle p_1, p_2 \rangle$ and the other which bears the same relationship to $\langle p'_1, p'_2 \rangle$. Any pair of similar triangles will suffice; however, the following two triangles are easy to construct.

Choose a point q_1 on the line $\{p_1, p_2\}$ and a point q'_1 on the line $\{p'_1, p'_2\}$ which satisfy the same *distance ratio* with respect to p_1 and p'_1, respectively. That is,

$$\frac{\text{length of } \{p_1, q_1\}}{\text{length of } \{p_1, p_2\}} = \frac{\text{length of } \{p'_1, q'_1\}}{\text{length of } \{p'_1, p'_2\}} = c_r \,. \tag{13}$$

For convenience, set $c_r = \frac{1}{2}$. That is, q_1 and q'_1 are respectively the midpoints of the lines $\{p_1, p_2\}$ and $\{p'_1, p'_2\}$. The coordinates for q_1 and q'_1 are given by

$$\left.\begin{aligned} q_1 &= \langle \tfrac{1}{2}[x(p_1)+x(p_2)],\ \tfrac{1}{2}[y(p_1)+y(p_2)] \rangle \,, \\ q'_1 &= \langle \tfrac{1}{2}[x(p'_1)+x(p'_2)],\ \tfrac{1}{2}[y(p'_1)+y(p'_2)] \rangle \,, \end{aligned}\right\} \tag{14}$$

where $x(p)$ and $y(p)$ denote the x-coordinates and y-coordinates of the point p.

Define the following coordinate differences:

$$\left.\begin{aligned} \Delta x &= x(q_1) - x(p_1) \,, & \Delta x' &= x(q'_1) - x(p'_1) \,, \\ \Delta y &= y(q_1) - y(p_1) \,, & \Delta y' &= y(q'_1) - y(p'_1) \,. \end{aligned}\right\} \tag{15}$$

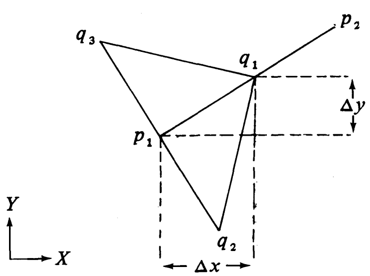
From which one obtains the points

$$\left.\begin{aligned} q_2 &= \langle x(p_1) + \Delta y,\ y(p_1) - \Delta x \rangle \,, & q_3 &= \langle x(p_1) - \Delta y,\ y(p_1) + \Delta x \rangle \,, \\ q'_2 &= \langle x(p'_1) + \Delta y',\ y(p'_1) - \Delta x' \rangle \,, & q'_3 &= \langle x(p'_1) - \Delta y',\ y(p'_1) + \Delta x' \rangle \,. \end{aligned}\right\} \tag{16}$$

It is easy to show that the triangles formed by the points q_1, q_2, q_3 and q'_1, q'_2, q'_3 are similar right-angled isosceles triangles. Moreover, the pair of lines $\{p_1, p_2\}$ and $\{q_2, q_3\}$ are mutually perpendicular and meet at p_1. Likewise, $\{p'_1, p'_2\}$ and $\{q'_2, q'_3\}$ are mutually perpendicular and intersect at p'_1. The lengths of the lines $\{q_1, q_2\}$ and $\{q_1, q_3\}$ are equal. So are the lengths of the lines $\{q'_1, q'_2\}$ and $\{q'_2, q'_3\}$. This construction is shown in figure 3.

The transformation τ must satisfy

$$\left.\begin{aligned} &q'_1 = \tau(q_1) \,, \\ &\text{and either} \\ &q'_2 = \tau(q_2) \quad \text{and} \quad q'_3 = \tau(q_3) \\ &\text{or} \\ &q'_2 = \tau(q_3) \quad \text{and} \quad q'_3 = \tau(q_2) \,. \end{aligned}\right\} \tag{17}$$



$$\begin{aligned} q_1 &= \langle \tfrac{1}{2}[x(p_1)+x(p_2)],\ \tfrac{1}{2}[y(p_1)+y(p_2)] \rangle \\ q_2 &= \langle x(p_1)+\Delta y,\ y(p_1)-\Delta x \rangle \\ q_3 &= \langle x(p_1)-\Delta y,\ y(p_1)+\Delta x \rangle \end{aligned}$$

Figure 3. The construction of the triangle formed by the points q_1, q_2, and q_3 with respect to the line $\{p_1, p_2\}$.

Therefore the mapping defined by the correspondence between a pair of distinct points $\langle p_1, p_2 \rangle$ associated with α and a pair of distinct points $\langle p'_1, p'_2 \rangle$ associated with γ yields two possible transformations τ_1^* and τ_2^*, for which $\tau_1^*(\alpha) \leq \gamma$ or $\tau_2^*(\alpha) \leq \gamma$ may hold. τ_1^* and τ_2^* are mirror reflections about the line $\{p_1, p_2\}$.

Finally, it is obvious that a correspondence between a single specified point p_1 associated with α and another such one p'_1 associated with γ does *not* determine a finite number of transformations τ for which $\tau(\alpha) \leq \gamma$ may hold. As before one can construct similar triangles relative to p_1 and p'_1, respectively. For convenience, let the triangle with respect to p_1 be an equilateral triangle centred at p_1. This triangle can be paired with an *infinite* number of equilateral triangles centred at p'_1 yielding an *infinite* number of transformations.

Problem 2: the generation of $T_{R,\gamma}$

The approach adopted is essentially to compute every *possible* transformation τ for membership in $T_{R,\gamma}$. Each τ in $T_{R,\gamma}$ represents a mapping between distinct points associated with α and an equal number of corresponding points associated with γ. By this is meant that τ takes each 'distinguishable' point of α to a corresponding 'distinguishable' point of γ. (The notion of a 'distinguishable' point of a shape will be made clear in the course of this section.) And, in particular, a pairing of just three of these 'distinguishable' points of α with three corresponding points of γ is sufficient to specify τ completely. This may be turned around, and each correspondence between triples of 'distinguishable' points of α and γ which form similar triangles may be claimed to define a possible transformation τ. Given this pairing, the method described earlier can be employed to determine the coefficients of a possible τ, which is then examined for membership in $T_{R,\gamma}$. That is, this computed τ is checked for the subshape relationship $\tau(\alpha) \leq \gamma$, in the following manner.

(a) For each labelled point $p{:}A$ of α, $\tau(p){:}A$ must be a labelled point of γ.

(b) For each maximal line $\{p_1, p_2\}$ in α, $\{\tau(p_1), \tau(p_2)\}$ must be contained in some maximal line in γ.

If τ satisfies this relationship, τ is in $T_{R,\gamma}$; otherwise, τ is rejected. An efficient algorithm for subshape determination is presented in Krishnamurti (1980).

Suppose τ is a transformation such that $\tau(a) \leq \gamma$. Then, it is known that if $p{:}A$ is a labelled point of α, $\tau(p){:}A$ is a labelled point of γ. Moreover, if p is a point of intersection of two maximal lines in α, $\tau(p)$ is a point of intersection of two maximal lines in γ. A *point of intersection* is the point at which two noncolinear maximal lines or their extensions meet. Clearly, two colinear maximal lines or their extension share an infinite number of common points, and two parallel maximal lines or their extensions never meet. Figure 4 presents examples of points of intersection.

At this stage, it is appropriate to remark that a point of intersection is really a disguised labelled point, since it reflects an aspect of the shape. Suppose the use of a special *intersection label*, denoted by, say, the symbol #, is permitted. Then, every

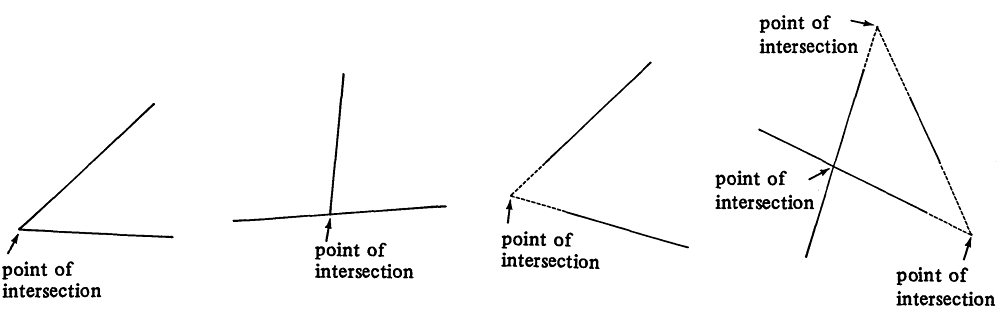


Figure 4. Examples of points of intersection.

point of intersection, $\hat{p}$ is essentially the labelled point, $\hat{p}$:#. Hence, for any labelled shape given by $\sigma = \langle s, P\rangle$, if $P_{\#}$ denotes the set of the points of intersection in σ, then σ can be represented by the ordered pair $\langle s, P+P_{\#}\rangle$. Therefore, it is not necessary to differentiate between a labelled point and a point of intersection. Henceforth, for convenience, they will be referred to simply as labelled points.

Consequently, a point p is a *distinguishable point* of a shape whenever for any transformation τ for which $\tau(\alpha) \leq \gamma$ holds, p and $\tau(p)$ are the same kind of points. For instance, if p is a point associated with the label A, $\tau(p)$ is a point associated with the same label A. Two distinguishable points, p_1 and p_2, are *distinct* if and only if they do not share the same coordinates. Hence, the initial choices for the distinguishable points of α and γ are from the sets of labelled points of α and γ.

In figure 5 there are two labelled shapes, viz α and γ; α consists of four maximal lines that form a square and a single labelled point of the form p:A situated at the centre of the square, and γ consists of six maximal lines that form a square divided into four squares. Each square, five in all, has at its centre, a labelled point of the form p':A. The labelled shape γ has five subshapes each of which is similar to α, and each of these subshapes can be obtained from a transformation that takes a triple of distinguishable points of α to a corresponding triple of distinguishable points of γ, the two triples of points forming similar triangles.

It may be noted, that each mapping between corresponding triples of distinct labelled points of α and γ which yields a transformation τ for which $\tau(\alpha) \leq \gamma$ holds,

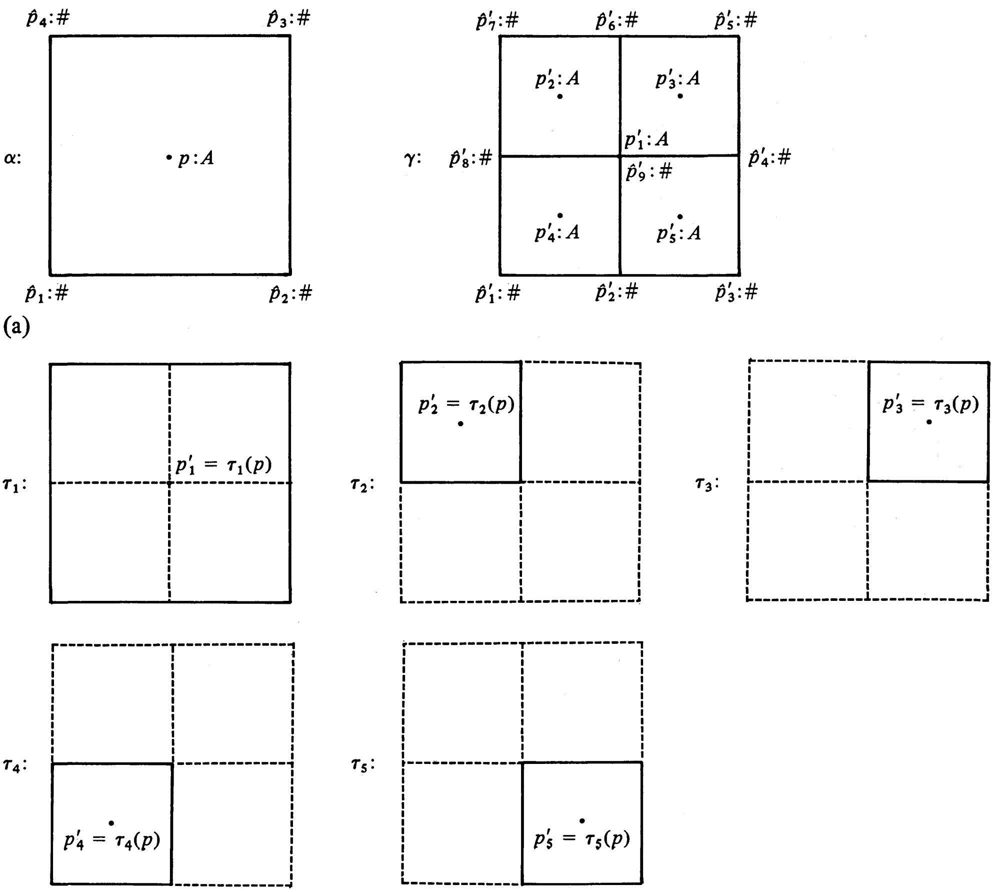


Figure 5. (a) Labelled shapes α and γ. (b) Transformations $\tau_1 - \tau_5$.

also determines the mapping between the remaining distinguishable points of α and an equal number of corresponding points associated with γ. Consequently, a triple of distinct labelled points of α can be *fixed*, and just the set of all possible mappings between this fixed triple of points associated with α and the corresponding points associated with γ such that each pairing of the triples of points form similar triangles be determined. Each mapping in this set describes a possible transformation which is, in turn, examined for membership in $T_{R,\gamma}$. Notice that in this case—that is, when α has at least three distinct labelled points which form a triangle—only a finite number of transformations are added to $T_{R,\gamma}$.

Now suppose α does not contain three labelled points which form a triangle. Then, it is still possible to determine candidate transformations that can be examined for membership in $T_{R,\gamma}$, provided one can find correspondence between pairs of distinguishable points of α and γ. In this case each correspondence yields, via the construction given by equations (14) through (17), two different pairings between three points associated with α and three points associated with γ, each of which defines a possible transformation.

Thus, the case when α contains at least two distinct labelled points all of which lie on a line poses no difficulty. These points are simply treated as the distinguishable points of α which pairwise can be mapped onto corresponding labelled points of γ to generate possible transformations to test for membership in $T_{R,\gamma}$. Moreover, using similar arguments as before, it is sufficient to consider just the mappings between a *fixed* pair of distinct labelled points of α and the corresponding labelled points of γ. Here, again, only a finite number of transformations are added to $T_{R,\gamma}$. Examples of α for which this case holds are shown in figure 6.

Suppose, instead, α contains fewer than two distinct labelled points. There are two cases to consider: α contains precisely one distinct labelled point; and α does not contain any labelled points. In the case when α contains a labelled point, there are three possible subcases for each of which the labelled point is a distinguishable point of γ. Two of these subcases are illustrated in figure 7 in which α consists of a single maximal line denoted by l and a single labelled point denoted by $p{:}A$. The third and last subcase will be considered later in this section.

Consider figure 7(a) in which the point p is not coincident with the maximal line l. For reasons that will become apparent, in general, p must not be coincident with any line colinear with l. Any transformation τ which satisfies $\tau(\alpha) \leq \gamma$ must preserve the relative disposition of the point p and the maximal line l. Notice that τ does not necessarily map l to a maximal line in γ. However, suppose a line, passing through p, is constructed at right angles to l such that this line and l intersect at a point denoted by $\perp(p)$. Term this point the *⊥-intersection point* of the labelled point p and the maximal line l. Notice that in this case $\perp(p) \neq p$. Then, clearly τ maps this point to a ⊥-intersection point of $\tau(p)$ in $\tau(\alpha)$. That is, τ maps the pair of points $\langle p, \perp(p)\rangle$ of α to a corresponding pair of points $\langle \tau(p), \perp[\tau(p)]\rangle$ of γ. In other words, the ⊥-intersection points of a shape are distinguishable points of the shape. Furthermore, since γ has a limited number of labelled points and since for each labelled point, p, there are at most as many $\perp(p)$ points as there are distinct partitions of γ into lists of colinear

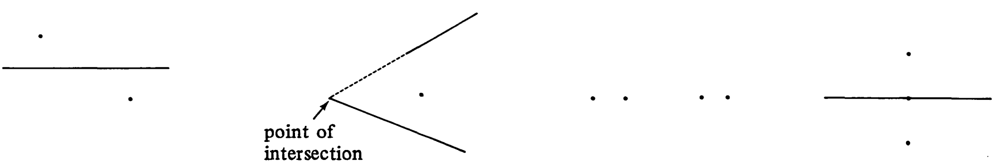


Figure 6. Examples of $s(\alpha)$ which consist of at least two distinguishable points all of which lie on a line. (• refer to labelled points.)

maximal lines, it follows that, in this case, only a limited number of possible transformations have to be examined for membership in $T_{R,\gamma}$, and $T_{R,\gamma}$ is finite.

On the other hand, consider figure 7(b) in which the labelled point p is coincident with the maximal line l. In fact, it is only necessary to ensure that p is coincident with a line colinear to l. That is, $\perp(p) = p$. In this situation there is an infinite number of transformations τ each of which satisfies $\tau(\alpha) \leq \gamma$. This is due to the fact that p and the end points of l do not form a triangle. Since τ needs only preserve the relative disposition of p and the end points of l, the end points of $\tau(l)$ correspond to points coincident with, but not necessarily equal to the end points of, a maximal line in γ. However, a finite number of transformations in $T_{R,\gamma}$ correspond to those generated by *forcing* two end points of maximal lines one in α and the other in γ as the second distinguishable points of α and γ, respectively. Notice that, under this restriction, only a finite number of transformations have to be examined for membership in $T_{R,\gamma}$.

Finally, consider the case when α contains no labelled points. In this case it is possible to determine some, but not all members of $T_{R,\gamma}$ by selecting one or more of the distinguishable points from the end points of maximal lines in the shapes. Figures 8, 9, and 10 illustrate this case. Consider figure 8 in which α comprises a single maximal line. For any labelled shape γ with at least one maximal line, α can be mapped onto any part of any maximal line in γ. Each such mapping yields a transformation τ for which $\tau(\alpha) \leq \gamma$ holds, and there is an infinite number of such transformations. Clearly, it is computationally impossible (and undesirable) to generate every possible transformation. Notice, however, that a finite number of these transformations correspond to mappings each of which takes the maximal line in α to a maximal line in γ.

Figure 9 presents two labelled shapes: α which consists of three colinear maximal lines l_1, l_2, and l_3, and γ which consists of a single maximal line l. Clearly, there is an infinite number of transformations τ which satisfy $\tau(\alpha) \leq \gamma$. However, there are two particular transformations τ_1^* and τ_2^* which are of interest: τ_1^* maps the tail of l_1 to the tail of l, and the head of l_3 to the head of l; every other end point of the

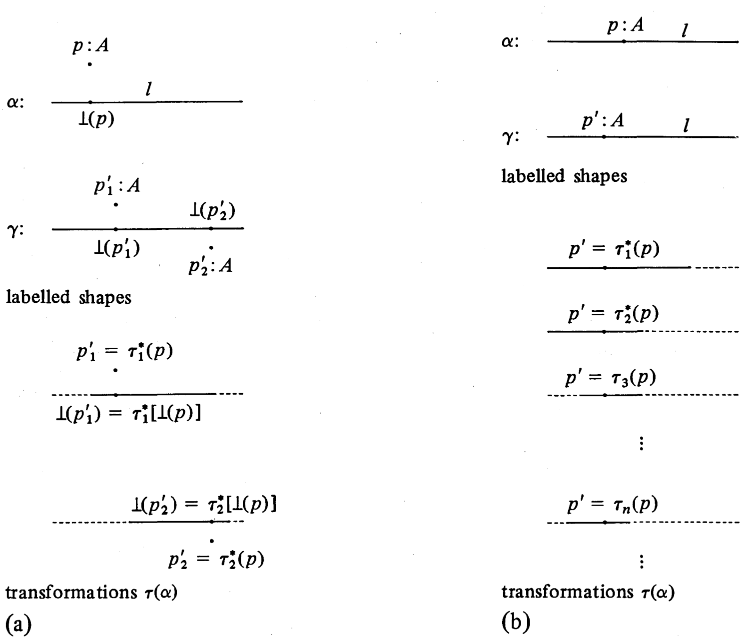


Figure 7. The case when α consists of exactly one distinct labelled point.

lines in α are mapped to points coincident with l; τ_2^* maps the tail of l_1 to the head of l, and the head of l_3 to the tail of l; every other end point of the lines in α are mapped to points coincident with l. Figure 9 represents a situation in which τ_1^* and τ_2^* are transformations which take a pair of distinct end points of maximal lines in α to a pair of distinct end points of maximal lines in γ.

Figure 10 illustrates examples of pathological shape rule situations which arise when α consists of at least two parallel maximal lines. In such situations, if the shape rule does apply, then there is either an infinite number of transformations τ for which $\tau(\alpha) \leq \gamma$ holds [figures 10(a) and (c)] or only a finite number of transformations τ for which $\tau(\alpha) \leq \gamma$ holds [figures 10(b) and (d)]. This is due to the fact τ must preserve in $\tau(\alpha)$ the relative lengths of the maximal lines of α as well as the relative distance between the two parallel lines. However, we may notice that in both cases above, a finite number of transformations can be obtained by mapping an end point of maximal lines in α to an end point of a maximal line in γ. In order to describe these

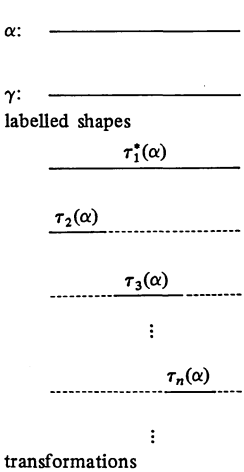


Figure 8. A shape rule situation that arises when α contains no labelled points.

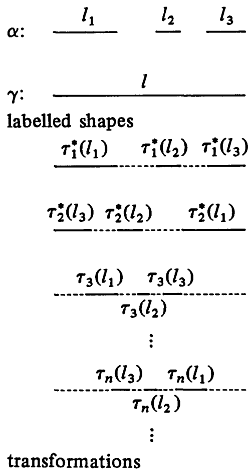


Figure 9. Another shape rule situation that arises when α contains no labelled points.

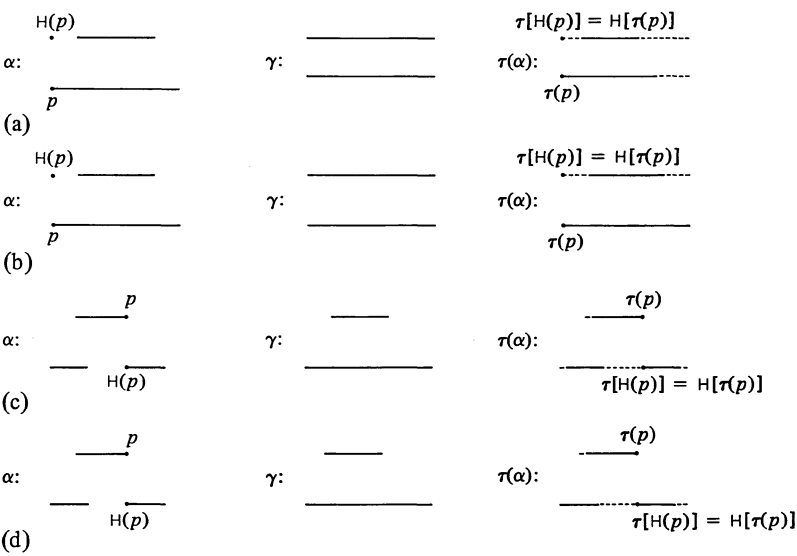


Figure 10. The case when α contains parallel maximal lines and no labelled points; (a) and (c) are for the infinite case, and (b) and (d) for the finite case.

transformations a second point is needed that is associated with α, and which maps onto a corresponding point associated with γ. This second point is determined as follows. Let l_1 and l_2 be two parallel maximal lines in α. Let p be an end point of, say l_1, which is mapped onto a corresponding end point $\tau(p)$ of a maximal line in γ. Suppose a line is constructed perpendicular to l_1 and passing through p. This line intersects l_2 at a point denoted by $\mathsf{H}(p)$. Term this point as the H-*intersection point* of the parallel maximal lines l_1 and l_2 with respect to the end point p of l_1. Then, clearly τ must map $\mathsf{H}(p)$ to a corresponding point $\mathsf{H}[\tau(p)]$ in $\tau(\alpha)$. That is, if an end point p of a maximal line is forced to be a distinguishable point of a shape, then all its $\mathsf{H}(p)$s are distinguishable points of the shape. Notice that in some cases $\mathsf{H}(p)$ may be an end point of a maximal line in α. Since there are only a finite number of maximal lines in γ, there are only a limited number of transformations that have to be examined for membership in $T_{R,\gamma}$. Notice that in the case when $T_{R,\gamma}$ is finite, all the members in $T_{R,\gamma}$ will be determined.

In each shape rule situation discussed above it has been assumed that α contains a sufficient number of points which may serve as distinguishable points of α. Suppose α has no maximal lines and precisely one distinct labelled point. Such shapes often occur as the left-hand side of shape rules in shape grammars when more than one initial shape is required. In these situations it is common practice to define a 'dummy' initial shape consisting of a single labelled point and to use shape rules to generate the initial shapes. In this case the *convention* will be used that $T_{R,\gamma}$ consists of translations each of which is given by the mapping of the labelled point of α to a corresponding labelled point of γ.

Therefore, shape rules can be divided into two classes: (a) those for which $T_{R,\gamma}$ is deterministic; and (b) those for which $T_{R,\gamma}$ is potentially infinite. For the latter class, a finite number of transformations can be obtained either by forcing as one or both distinguishable points of α and γ end points of maximal lines in α and γ, or by introducing a convention as to the nature of the transformations in $T_{R,\gamma}$. In order to classify the shape rules provisos are stipulated which regulate the application of shape rules.

Provisos to govern shape rule application

Let R, $R = \langle\alpha, \alpha-\beta, \beta-\alpha\rangle$, be a shape rule. Let γ be the labelled shape to which R is applied. Let, for any labelled shape σ, $s(\sigma)$ denote its shape and $P(\sigma)$ denote its associated set of labelled points. That is, σ is represented by the pair, $\langle s(\sigma), P(\sigma)\rangle$. For a labelled shape σ, $\sigma = \langle s(\sigma), P(\sigma)\rangle$, define the following sets:

(a) $P^*(\sigma) = P(\sigma)+P_{\#}(\sigma)$ where $P_{\#}(\sigma)$ denotes the set of points of intersection of maximal lines in $s(\sigma)$.

(b) $P_{\bullet}(\sigma) = \{p | p$ is an end point of a maximal line in $s(\sigma)\}$.

(c) $P_{\perp}(\sigma) = \{\perp(p) \neq p | p \in P(\sigma)\}$.

(d) $P_{\mathsf{H}}(\sigma) = \{\mathsf{H}(p) | p \in P_{\bullet}(\sigma)\}$.

Proviso 1: $P^*(\alpha)$ contains at least three distinct points, say p_1, p_2, and p_3 which do not all lie on a line.

R applies to γ whenever there is a transformation τ which maps $\langle p_1, p_2, p_3\rangle$ to corresponding points $\langle p'_1, p'_2, p'_3\rangle$ in $P^*(\gamma)$ which do not all lie in a line such that $\tau(\alpha) \leq \gamma$. In this case, $T_{R,\gamma}$ is finite.

Proviso 2: $P^*(\alpha)$ contains at least two distinct points, say p_1 and p_2, and *every point in $P^*(\alpha)$ lies on a line.*

R applies to γ whenever there is a transformation τ which maps $\langle p_1, p_2\rangle$ to corresponding points $\langle p'_1, p'_2\rangle$ in $P^*(\gamma)$ such that $\tau(\alpha) \leq \gamma$. Here, again, $T_{R,\gamma}$ is finite.

Proviso 3: $P^*(\alpha)$ contains precisely one distinct point, say p_1, and *$P_{\perp}(\alpha)$ contains at least one point, say $\perp^*(p_1)$.*

R applies to γ whenever there is a transformation τ which maps $\langle p_1, \perp^*(p_1)\rangle$ to corresponding points $\langle p'_1, \perp(p'_1)\rangle$ where $p'_1 \in P^*(\gamma)$ and $\perp(p'_1) \in P_\perp(\gamma)$ such that $\tau(\alpha) \leq \gamma$. Here, again, $T_{R,\gamma}$ is finite.
Proviso 4: $P^*(\alpha)$ contains exactly one distinct labelled point, say p_1, and $P_\perp(\alpha) = \emptyset$.
R applies to γ whenever there is a transformation τ such that when $s(\alpha)$ is nonempty τ maps the pair of points $\langle p_1, p_2\rangle$, $p_2 \in P_\bullet(\alpha)$, onto the corresponding points $\langle p'_1, p'_2\rangle$, $p'_1 \in P^*(\gamma)$, $p'_2 \in P_\bullet(\gamma)$, with $\tau(\alpha) \leq \gamma$; and when $s(\alpha)$ is empty, τ is a translation which maps p_1 onto a corresponding point p'_1 in $P^*(\gamma)$.
Proviso 5: $P^*(\alpha)$ is empty.
R applies to γ wherever there is a transformation τ with $\tau(\alpha) \leq \gamma$ such that when $s(\alpha)$ consists of at least two parallel maximal lines τ maps $\langle p_1, \mathrm{H}(p_1)\rangle$, $p_1 \in P_\bullet(\alpha)$, $\mathrm{H}(p_1) \in P_\mathrm{H}(\alpha)$ to the corresponding points $\langle p'_1, \mathrm{H}(p'_1)\rangle$, where $p'_1 \in P_\bullet(\gamma)$ and $\mathrm{H}(p'_1) \in P_\mathrm{H}(\gamma)$; and for $s(\alpha)$ otherwise maps $\langle p_1, p_2\rangle$, $p_1, p_2 \in P_\bullet(\alpha)$ to corresponding points $\langle p'_1, p'_2\rangle$, where $p'_1, p'_2 \in P_\bullet(\gamma)$.

The conditions of precisely one of these provisos applies to any shape rule with $\alpha \neq \langle s_\phi, \emptyset\rangle$. Provisos 1, 2, and 3 are the normal situations in the sense that if γ contains a subshape similar to α, then the transformation which defines the similarity relationship is determinable by one of these provisos. Provisos 4 and 5 are restrictions to the normal situations in the sense that γ may contain subshapes similar to α not determinable by any of the provisos. Proviso 4 includes the convention for handling shape rule situations occurring in shape grammars requiring more than one initial shape.

Soundness of the provisos
Suppose τ is a transformation for which the subshape relation $\tau(\alpha) \leq \gamma$ holds. Then, there exists at least one transformation τ^* which can be determined by the provisos such that $\tau^*(\alpha) \leq \gamma$ *and* either $\tau^* = \tau$ or τ^* is easily constructed from τ. Note that τ maps each, if any, distinguishable point of α to a corresponding distinguishable point of γ. Note also that α satisfies the conditions for precisely one of the provisos. There are three cases to consider.
(1) *α satisfies the conditions for provisos 1, 2, or 3*
Since the provisos examine every possible correspondence between the distinguishable points of α and γ, τ will be one of the transformations so examined.
(2) *α satisfies the condition for proviso 4*
Let p denote the distinct labelled point or point of intersection in α. The case when $s(\alpha)$ is the empty shape s_ϕ is trivially true, since for if τ is not a translation, there is a translation τ^* which maps p onto $\tau(p)$ and which can be constructed from τ by the composition $\tau^*[\tau^{-1}(\tau)]$ where τ^{-1} is the inverse transformation of τ. Therefore, it may be supposed that $s(\alpha)$ is nonempty. There are two possibilities. Either τ maps at least one end point of a maximal line in α to a corresponding end point of a maximal line in γ in which case τ is the desired transformation, or τ maps every point in $P_\bullet(\alpha)$ to points coincident with, but not equal to the end points of, maximal lines in γ. Notice that for any scale transformation, τ_c, $\tau_c[\tau(\alpha)]$ is similar to $\tau(\alpha)$. However, τ_c maps $\tau(p)$ to a point $\tau_c[\tau(p)]$ which is not identical to $\tau(p)$. Nevertheless, there is a specific translation τ_t which maps $\tau_c[\tau(p)]$ to $\tau(p)$; that is, $\tau_t\{\tau_c[\tau(p)]\} = \tau(p)$. Moreover, the resulting labelled shape $\tau_t\{\tau_c[\tau(\alpha)]\}$ has the following property. For each maximal line l in α, the transformations of the line, $\tau_t\{\tau_c[\tau(l)]\}$ and $\tau(l)$ are colinear. This follows from the fact that $P_\perp(\alpha) = \emptyset$. Hence, either $\tau_t\{\tau_c[\tau(\alpha)]\} \leq \gamma$ or $\tau_t\{\tau_c[\tau(\alpha)]\} \nleq \gamma$. In the former case, if $\tau_t[\tau_c(\tau)]$ maps at least one point in $P_\bullet(\alpha)$ to a corresponding end point in $P_\bullet(\gamma)$, then the composition $\tau_t[\tau_c(\tau)]$ is the desired transformation τ^*. Otherwise, τ_c has scaled $\tau(\alpha)$ by too low a factor. Hence, by *enlarging* $\tau_t\{\tau_c[\tau(\alpha)]\}$ by a scale factor c, $c > 1$, followed by the appropriate trn translation, a new transformation is derived, say τ', for which $\tau'(\alpha) \leq \gamma$ may hold.

In the latter case, τ_c has scaled $\tau(\alpha)$ by too large a factor. Hence, by *shrinking* $\tau_t\{\tau_c[\tau(\alpha)]\}$ by a scale factor c, $0 < c < 1$, followed by the appropriate translation, a new transformation is obtained, say τ'', for which $\tau''(\alpha) \leq \gamma$ may hold. In either case, it is clear that there is an alternating sequence of scale transformations and translations, the composition of which yields, say $\hat{\tau}$, which satisfies $\tau^* = \hat{\tau}(\tau)$ and $\tau^*(\alpha) \leq \gamma$.

(3) *α satisfies the condition for proviso 5*

The case when $s(\alpha)$ has no parallel maximal lines is trivially true, since if τ is not the desired transformation, then τ^* can be constructed from τ by applying to τ a sequence of transformations which consists of a translation, a scale and another translation the details of which is left to the reader. As an example, suppose α has exactly one maximal line l. Then, $\tau(l)$ must be contained in some maximal line l' in γ. First, apply the translation τ_{t_1} which maps the midpoint of $\tau(l)$ to the midpoint of l'. Next apply the scale transformation τ_c by the scale factor, c, which equals the ratio of the length of l' to the length of $\tau_{t_1}[\tau(l)]$. Last, apply the translation τ_{t_2} which maps the midpoint of $\tau_c\{\tau_{t_1}[\tau(l)]\}$ to the midpoint of l'. Clearly, $\tau^* = \tau_{t_2}\{\tau_c[\tau_{t_1}(\tau)]\}$ gives the desired transformation. Suppose $s(\alpha)$ has at least two parallel maximal lines. There are two possibilities: either τ maps an end point, say p, in $P_\bullet(\alpha)$ to a corresponding end point, say p', in $P_\bullet(\gamma)$, or τ maps each end point in $P_\bullet(\alpha)$ to points coincident with, but not equal to the end points of, maximal lines in γ. In the former case, τ is the desired transformation since τ will also map the H-intersection point(s) of p, $\mathrm{H}(p)$, to corresponding H-intersection point(s) of p', $\mathrm{H}(p')$, $\mathrm{H}(p') = \tau[\mathrm{H}(p)]$. In the latter case, notice that any translation τ_t of $\tau(\alpha)$ preserves the euclidean distance between any pair of parallel maximal lines in $\tau(\alpha)$. For each end point $\tau(p)$ of a maximal line $\tau(l)$ in $\tau(\alpha)$, let p' be the end point of a maximal line l' in γ *nearest* to it [in the euclidean sense] such that l' is colinear with $\tau(l)$. [In fact, it is possible to show that $\tau(l)$ is contained in l'.] Let $d[p', \tau(p)]$ be the euclidean distance between p' and $\tau(p)$. Let, over all such pairs, the pair, say $\langle p'_1, \tau(p_1)\rangle$ have the minimum distance $d[p'_1, \tau(p_1)]$. Let τ_t be the translation which maps $\tau(p_1)$ to p'_1. Clearly, $\tau_t[\tau(\alpha)] \leq \gamma$. Moreover, $\tau_t(\tau)$ is the desired transformation τ^*.

When $T_{R,\gamma} = \emptyset$, none of the possible transformations generated by the provisos will satisfy the subshape relationship.

Thus, for any left-hand side α of a shape rule there is a proviso which will always yield a transformation τ^* for which $\tau^*(\alpha) \leq \gamma$ holds if and only if $T_{R,\gamma}$ is nonempty. If, in addition, $T_{R,\gamma}$ is finite the proviso will determine every transformation in $T_{R,\gamma}$.

The number of possible transformations

The conditions imposed on the provisos define the type of the shape rule. If the left side of a shape rule satisfies the condition for proviso j, $1 \leqslant j \leqslant 5$, the shape rule is said to be of *type j*. For each type of shape rule there is only a limited number of possible transformations that have to be examined for membership in $T_{R,\gamma}$.

Suppose α contains the labels (including #) $A_1, A_2, ..., A_{N_\alpha}$. Let n_j, $1 \leqslant j \leqslant N_\alpha$, denote the number of labelled points of γ having the form $p:A_j$. Moreover, let $n_1 \leqslant n_2 \leqslant ... \leqslant n_{N_\alpha}$. Let $n_\bullet(\alpha)$ and $n_\bullet(\gamma)$ denote respectively the cardinalities of $P_\bullet(\alpha)$ and $P_\bullet(\gamma)$. Let M_α and M_γ be the number of disjoint subshapes of $s(\alpha)$ and $s(\gamma)$ wherein each subshape consists of colinear maximal lines. Let $m(\alpha)$ and $m(\gamma)$ denote respectively the number of maximal lines in $s(\alpha)$ and $s(\gamma)$. Let N_τ be the number of possible transformations that have to be examined for membership in $T_{R,\gamma}$.

Consider shape rules of types 1 and 2. It may be supposed that for shape rules of type 1 the choice for the fixed triple of points is from one of the four combinations ranging from $\langle p_1{:}A_1, p_2{:}A_1, p_3{:}A_1\rangle$ through $\langle p_1{:}A_1, p_2{:}A_2, p_3{:}A_3\rangle$; and for shape rules of type 2 the fixed pair is either $\langle p_1{:}A_1, p_2{:}A_1\rangle$ or $\langle p_1{:}A_1, p_2{:}A_2\rangle$. The fixed set of labelled points so chosen yields the smallest possible N_τ.

At this stage, the remark is made that shape rules of type 1 are equivalent to those of type 2 in the following sense: any transformation that maps a triple of labelled points of α to a corresponding triple of labelled points of $\tau(\alpha)$ in γ, also maps any triangle relative to α to a similar triangle which bears the same relationship to $\tau(\alpha)$. And so, for shape rules of type 1 discard the third point p_3 from the triple and employ the construction for the triples provided by equations (14) through (17) to specify τ. In fact, it is sometimes computationally more efficient to do so. For instance, suppose the fixed triple is $\langle p_1{:}A_1, p_2{:}A_1, p_3{:}A_1\rangle$. Then, by means of proviso 1, the number of possible mappings that have to be examined is $n_1(n_1-1)(n_1-2)$. On the other hand, by means of proviso 2, the number of possible mappings that have to be examined is $n_1(n_1-1)2$. Clearly, proviso 2 is more efficient when $n_1 \geqslant 5$. Below are given the least conditions on the n_js under which it is computationally more efficient to treat a shape rule of type 1 as though it was of type 2:

fixed triple	least condition on the n_js
$\langle p_1{:}A_1, p_2{:}A_1, p_3{:}A_1\rangle$	$n_1 \geqslant 5$
$\langle p_1{:}A_1, p_2{:}A_1, p_3{:}A_2\rangle$	$n_2 \geqslant 3$
$\langle p_1{:}A_1, p_2{:}A_2, p_3{:}A_2\rangle$	$n_2 \geqslant 4$
$\langle p_1{:}A_1, p_2{:}A_2, p_3{:}A_3\rangle$	$n_3 \geqslant 3$

The following shapes, shown in figure 11, are of particular interest:
(a) a single maximal line;
(b) a set of colinear maximal lines;
(c) parallel sets of colinear maximal lines—that is, each maximal line in a set is parallel to any other maximal line in any other set;
(d) two maximal lines which intersect;
(e) two or more sets of colinear maximal lines such that each line in a set intersects every other line in any other set at the same point.

For shape rules of types 3 and 5 $s(\alpha)$ is identical to one of the shapes: (a), (b), or (c); whereas for shape rules of type 4, when $s(\alpha)$ is not empty, $s(\alpha)$ is identical to one of the shapes (a), (b), (d), or (e).

As in the case of shape rules of types 1 or 2, for shape rules of type 3 it is only necessary to consider all possible mappings between a *fixed* pair of distinguishable points of α and corresponding points associated with α. Clearly, in this case N_τ is minimized if the fixed labelled point of α is of the form $p_1{:}A_1$.

Consider shape rules of type 4 with a nonempty $s(\alpha)$. Clearly, in this case the distinct labelled point or point of intersection is fixed, and it may be assumed to take the form $p_1{:}A_1$. Moreover, since $P_\perp(\alpha) = \emptyset$, p_1 lies on a line colinear with *every* maximal line in $s(\alpha)$, and, hence, it is only necessary to consider the correspondences

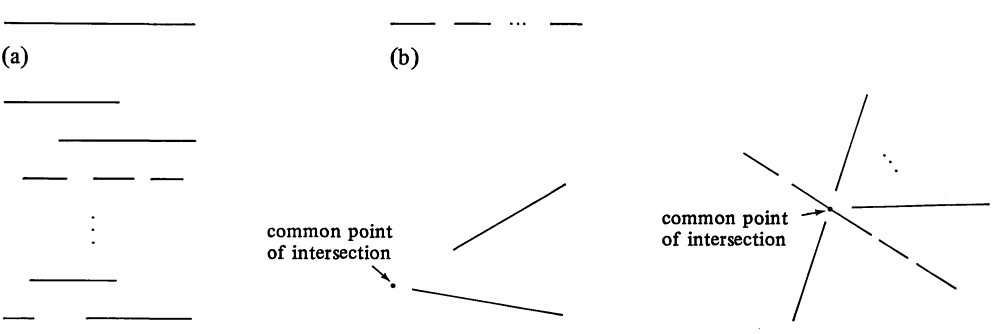


Figure 11. Examples of the possible shapes $s(\alpha)$ that may occur in shape rules of types 3, 4, and 5.

between $\langle p_1, p_2\rangle$, p_2 is an end point of a maximal line in $s(\alpha)$, and $\langle p'_1, p'_2\rangle$ where p'_2 is an end point of a maximal line in $s(\gamma)$ which is colinear with the line with which p'_1 is coincident. p'_1, of course, is the point associated with γ corresponding to p_1.

Finally, consider shape rules of type 5, for which, clearly, the M_α subshapes are all parallel to each other. Let τ be a transformation such that $\tau(\alpha) \leq \gamma$. Then, τ must map each of the M_α subshapes onto either a single maximal line or a collection of colinear maximal lines. There are two cases to consider: $M_\alpha = 1$ and $M_\alpha > 1$.

In the case when $M_\alpha = 1$, there must be—vide proviso 5—two end points of maximal lines in $s(\alpha)$, p_1 and p_2, such that $\tau(p_1)$ and $\tau(p_2)$ are end points of maximal lines in $s(\gamma)$. Moreover, if $\{p_1, p_2\}$ is a maximal line, $\{\tau(p_1), \tau(p_2)\}$ is also a maximal line. In order to determine the possible transformations, it is sufficient to consider the mappings between $\langle p_1, p_2\rangle$ and corresponding points $\langle p'_1, p'_2\rangle$ such that the line $\{p'_1, p'_2\}$ is colinear with some maximal line in $s(\gamma)$. That is, one need only consider the correspondence between $\langle p_1, p_2\rangle$ and the end points of maximal lines in each of the M_α subshapes in turn.

Each correspondence yields two possible transformations τ_1^* and τ_2^* which are mirror reflections of each other. Notice that if $\tau_1^*(\alpha) \leq \gamma$, $\tau_2^*(\alpha) \leq \gamma$ holds. Moreover, $\tau_1^*(\alpha) = \tau_2^*(\alpha)$. However, $\tau_1^*(\beta)$ is not necessarily identical to $\tau_2^*(\beta)$. Since each correspondence is a mapping of a pair of end points $\langle p_1, p_2\rangle$ onto a corresponding pair of end points $\langle p'_1, p'_2\rangle$, the computation can be speeded up by considering just the set of all pairings between pairs of end points $\{p_1, p_2\}$ and $\{p'_1, p'_2\}$. Each pairing yields two possible correspondences: $\langle p_1, p_2\rangle$ maps onto $\langle p'_1, p'_2\rangle$ and $\langle p_1, p_2\rangle$ maps onto $\langle p'_2, p'_1\rangle$. In other words, one can fix the ordering on the points p_1 and p_2 and on the points p'_1 and p'_2. Therefore, it may be supposed that $p_1 < p_2$ and $p'_1 < p'_2$. There are two cases to consider:

(1) *p_1 is the tail (head) of a maximal line and p_2 is the head (tail) of a maximal line.* Points p'_1 and p'_2 are respectively the tail (head) and head (tail) of maximal lines. In this case both the correspondences $\langle p_1, p_2\rangle$ maps onto $\langle p'_1, p'_2\rangle$ and $\langle p_1, p_2\rangle$ maps onto $\langle p'_2, p'_1\rangle$ may yield transformations τ for which $\tau(\alpha) \leq \gamma$ holds [see figures 12(a) and 12(b)].

(2) *p_1 and p_2 are both tails (heads) of maximal lines.*
In this case p'_1 and p'_2 are either both tails or both heads of maximal lines. In the case that p_1 and p'_1 are both tails or both heads of maximal lines, only the correspondence $\langle p_1, p_2\rangle$ may yield transformations τ for which $\tau(\alpha) \leq \gamma$ holds [figure 12(c)]; and for p_1 and p'_1 otherwise, only the correspondence $\langle p_1, p_2\rangle$ maps onto $\langle p'_2, p'_1\rangle$ needs to be considered [figure 12(d)].

A probabilistic argument is now given to estimate the average for N_τ, denoted by $\hat{N}_\tau$. Each pair of maximal lines that share a common end point cannot be colinear. Hence $\frac{1}{2}n.(\gamma) \leqslant m(\gamma) \leqslant \lfloor \frac{1}{2}[n.(\gamma)+M_\gamma] \rfloor$, where for any real number r, $\lfloor r \rfloor$ denotes the greatest integer not exceeding r. The assumption that the maximal lines are equally distributed among the M_γ subshapes of $s(\gamma)$, gives

$$\hat{N}_\tau = 4m(\alpha)m(\gamma) + m(\alpha)[m(\alpha)-1]m(\gamma)\left\lceil \frac{3m(\gamma)}{M_\gamma} - 1 \right\rceil .$$

In the case when $M_\alpha > 1$, any transformation τ for which $\tau(\alpha) \leq \gamma$ must map—vide proviso 5—an end point p_1 of a maximal line in $s(\alpha)$ to a corresponding end point p'_1 of a maximal line in $s(\gamma)$. Moreover, τ must map each of $(M_\alpha - 1)$ H-intersection points of p_1 to a corresponding H-intersection point of p'_1. That is, for each p'_1 there must be at least $(M_\alpha - 1)$ H-intersection points. Clearly, τ must map each of the M_α subshapes of $s(\alpha)$ into a distinct subshape of $s(\gamma)$. That is, τ must map each collection of colinear maximal lines in $s(\alpha)$ into a collection of colinear maximal lines in $s(\gamma)$.

Each mapping between corresponding pairs of points $\langle p_1, \mathsf{H}(p_1)\rangle$ and $\langle p'_1, \mathsf{H}(p'_1)\rangle$ yields two possible transformations, say τ^*_1 and τ^*_2. Let τ^*_1 preserve in $\tau^*_1(\alpha)$, the original ordering on the maximal lines in α. Since τ^*_2 is a mirror reflection of τ^*_1, τ^*_2 reverses this ordering of maximal lines in $\tau^*_2(\alpha)$. Let p be an end point of a maximal line such that $\{p, p_1\}$ is colinear with this maximal line. If $p < p_1$, then $\tau^*_1(p) < \tau^*_1(p_1)$ and $\tau^*_2(p_1) < \tau^*_2(p)$, and if $p_1 < p$, then $\tau^*_1(p_1) < \tau^*_1(p)$ and $\tau^*_2(p) < \tau^*_2(p_1)$. Clearly, if $\tau^*_1(\alpha) \leq \gamma$, then $\tau^*_2(\alpha) \not\leq \gamma$, and if $\tau^*_2(\alpha) \leq \gamma$, then $\tau^*_1(\alpha) \not\leq \gamma$. That is, at most one of τ^*_1 and τ^*_2 will satisfy the subshape relationship. There are two cases to consider:

(1) *p_1 is the tail (head) of a maximal line in $s(\alpha)$ and p'_1 is tail (head) of a maximal line in $s(\gamma)$.*

In this case, only τ^*_1 may possibly satisfy $\tau^*_1(\alpha) \leq \gamma$.

(2) *p_1 is the tail (head) of a maximal line in $s(\alpha)$ and p'_1 is the head (tail) of a maximal line in $s(\gamma)$.*

In this case only τ^*_2 may possibly satisfy $\tau^*_2(\alpha) \leq \gamma$.

Now, consider a possible transformation τ_1 which maps $\langle p_1, \mathsf{H}(p_1)\rangle$ to a corresponding $\langle p'_1, \mathsf{H}(p'_1)\rangle$. Another possible transformation τ_2 is determined when $\langle p_1, \mathsf{H}(p_1)\rangle$ is

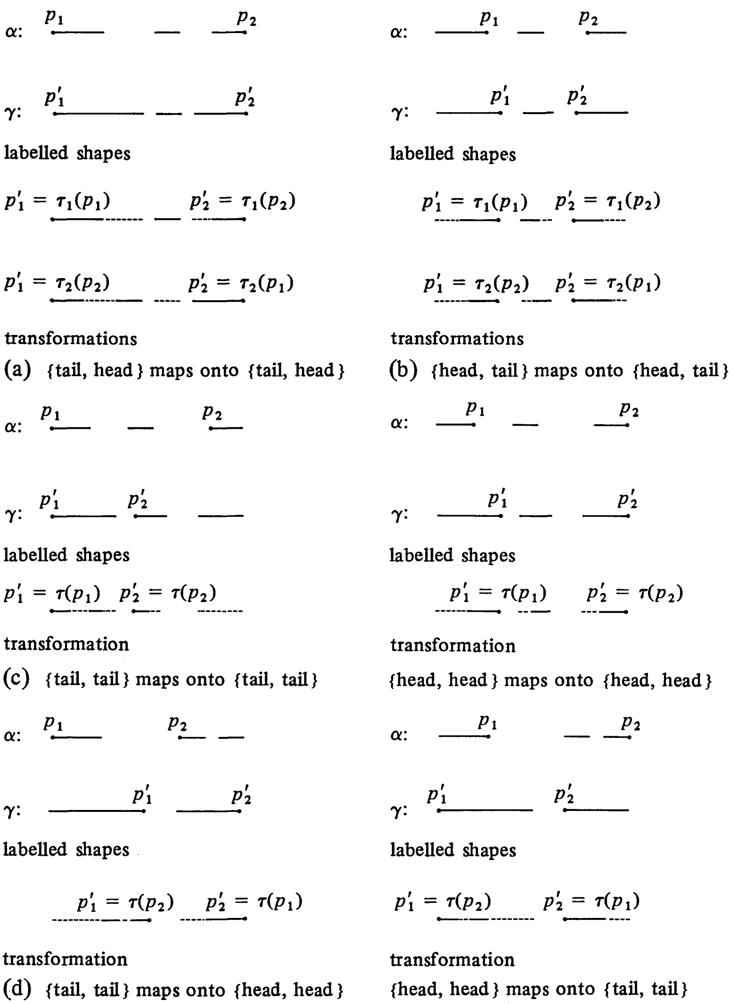


Figure 12. The possible correspondences between pairs of points for shape rules of type 5 with $M_\alpha = 1$.

mapped onto a corresponding $\langle p_2', \mathrm{H}(p_2')\rangle$, provided there is a translation τ' such that $\tau'(p_1') = p_2'$ and $\tau'[\mathrm{H}(p_1')] = \mathrm{H}(p_2')$. Then, τ_2 is given by the composition $\tau'(\tau_1)$. On the other hand a possible transformation τ_3 is determined when $\langle p_2, \mathrm{H}(p_2)\rangle$ is mapped onto $\langle p_1', \mathrm{H}(p_1')\rangle$, provided there is a translation τ'' such that $\tau''(p_2) = p_1$ and $\tau''[\mathrm{H}(p_2)] = \mathrm{H}(p_1)$. In this case, τ_3 is given by the composition $\tau_1(\tau'')$. Consequently, only a limited number of possible transformations have to be computed. The remaining transformations are obtained by simple translations of these basic transformations. There are at most $M_\alpha M_\gamma(M_\gamma - 1)$ such basic transformations. Suppose the maximal lines in $s(\alpha)$ and $s(\gamma)$ are uniformly distributed among their respective M_α and M_γ subshapes. Then, one has $\hat{N}_\tau \simeq [4m(\alpha)m(\gamma)/(M_\alpha M_\gamma)]\mathrm{O}[M_\alpha M_\gamma(M_\gamma - 1)]$, where O is the order function.

Finally, since each maximal line has two end points, $n_*(\alpha) = \mathrm{O}[2m(\alpha)]$ and $n_*(\gamma) = \mathrm{O}[2m(\gamma)]$. Therefore, it follows from the preceding arguments that N_τ satisfies one of the following inequalities:

for type 1, $\quad n_1(n_1-1)(n_1-2) \leqslant N_\tau \leqslant n_1 n_2 n_3$;

for type 2, $\quad 2n_1(n_1-1) \leqslant N_\tau \leqslant 2n_1 n_2$;

for type 3, $\quad N_\tau \leqslant 2n_1 M_\gamma$;

for type 4, $$N_\tau \begin{cases} = n_1 , & s(\alpha) = s_\phi , \\ \leqslant 8n_1\mathrm{O}[m(\alpha)m(\gamma)] , & \text{otherwise}; \end{cases}$$

for type 5, $$\hat{N}_\tau \simeq \begin{cases} 4m(\alpha)m(\gamma) + \mathrm{O}\{[m(\alpha)m(\gamma)]^2/M_\gamma\} , & M_\alpha = 1 , \\ 4\mathrm{O}[m(\alpha)m(\gamma)M_\gamma] , & \text{otherwise.} \end{cases}$$

The shape rule application algorithm

An algorithm is now presented, based on the ideas developed in the preceding section, to construct the set $T_{R,\gamma}$, the set of all transformations under which the shape rule R applies to the current shape γ, subject to the restrictions imposed by the provisos. This section is divided into two parts. First, the internal representations for shapes and shape rules are described. Second, a step by step description of the algorithm is presented. Wherever necessary illustrations are provided to facilitate the description. Although no formal proof for the algorithm is given, sufficient informal arguments are provided in the description to demonstrate its correctness.

Internal representation for shapes and shape rules

Let σ be a labelled shape, let $s(\sigma)$ denote the set of maximal lines in σ, and let $P(\sigma) + P_\#(\sigma)$ denote the set of labelled points and points of intersection in σ. $P(\sigma) + P_\#(\sigma)$ is partitioned into subsets $P_1(\sigma)$, $P_2(\sigma)$, ..., $P_{N_\sigma}(\sigma)$, where each $P_j(\sigma)$, $1 \leqslant j \leqslant N_\sigma$, consists of all the points in σ having the label $A_j(\sigma)$. The labels satisfy $A_1 < ... < A_{N_\sigma}$. For each j, $1 \leqslant j \leqslant N_\sigma$, let the cardinality of $P_j(\sigma)$ be denoted by $n_j(\sigma)$.

The shape $s(\sigma)$ is partitioned into disjoint subshapes $s_1(\sigma), s_2(\sigma), ..., s_{M_\sigma}(\sigma)$ each of which consists of colinear maximal lines. For all $i \neq k$, $1 \leqslant i, k \leqslant M_\sigma$, the shape intersection $s_i(\sigma) \cdot s_k(\sigma)$ is the empty shape s_ϕ. The number of maximal lines in $s_k(\sigma)$, $1 \leqslant k \leqslant M_\sigma$, is denoted by $m_k(\sigma)$. The line descriptor for the maximal lines in $s_k(\sigma)$, $1 \leqslant k \leqslant M_\sigma$, is denoted by $\psi_k(\sigma) = \langle \mu_k(\sigma), \nu_k(\sigma)\rangle$, and satisfies $\psi_1(\sigma) < ... < \psi_{M_\sigma}(\sigma)$.

The *current* labelled shape γ is represented internally as the ordered triple, $\gamma = \langle s(\gamma), P^*(\gamma), \Pi(\gamma)\rangle$, where $P^*(\gamma) = P(\gamma) + P_\#(\gamma)$, and $\Pi(\gamma) = \langle \pi(1), ..., \pi(N_\gamma)\rangle$ gives a permutation of the indices of $P_1(\gamma)$, ..., $P_{N_\gamma}(\gamma)$ satisfying $n_{\pi(1)}(\gamma) < ... < n_{\pi(N_\gamma)}(\gamma)$. $\pi(j)$ is the index of the jth largest set $P_{\pi(j)}(\gamma)$.

Each shape rule, R: $\alpha \rightarrow \beta$, is represented internally by the four-tuple R, with $R = \langle \alpha, \alpha-\beta, \beta-\alpha, type \rangle$, and by an associated set DP_α, where

α is the labelled shape $\langle s(\alpha), P^*(\alpha) \rangle$ with

$$P^*(\alpha) = \begin{cases} P(\alpha)+P_\#(\alpha)+P_\perp(\alpha), & \text{for } type = 3, \\ P(\alpha)+P_\#(\alpha)+P_.(\alpha), & \text{for } type = 4, \\ P(\alpha)+P_\#(\alpha), & \text{otherwise;} \end{cases}$$

$\alpha-\beta$ is the labelled shape $\langle s(\alpha-\beta), P(\alpha-\beta) \rangle$;

$\beta-\alpha$ is the labelled shape $\langle s(\beta-\alpha), P(\beta-\alpha) \rangle$;

type determines the proviso under which the shape rule applies and takes on values from 1 through 5, that is, the type of shape rule;

DP_α is a set of distinguishable points associated with α and is, initially, given by:
(a) for $type = 1$, a triple of distinct points in $P(\alpha)+P_\#(\alpha)$ which form a triangle,
(b) for $type = 2$, a pair of distinct points in $P(\alpha)+P_\#(\alpha)$,
(c) for $type = 3$, the distinct point in $P(\alpha)$ and a point in $P_\perp(\alpha)$
(d) for $type = 4$, the distinct point in $P(\alpha)+P_\#(\alpha)$;
(e) for $type = 5$, the empty set.

Notice that DP_α initially contains the fixed distinguishable points associated with α. For computational reasons, in order to minimize the number of transformations that have to be examined, these points in DP_α are computed once for each application of the shape rule.

Algorithm: the construction of $T_{R,\gamma}$
The algorithm comprises five steps:
0. initialization;
1. the construction of the fixed distinguishable points in DP_α;
2. the mappings for shape rules of types 1, 2, 3, and 4;
3. the mappings for shape rules of type 5;
4. finishing touches.

Step 0: Initialization
Substep 0.1: Let nt be the number of transformations in $T_{R,\gamma}$, and f the number of fixed labelled points in DP_α; then

$$nt \leftarrow 0, \quad f \leftarrow \begin{cases} 4\text{-}type, & \text{for } type \leqslant 3, \\ 5\text{-}type, & \text{otherwise.} \end{cases}$$

If $f = 0$, go to *substep 3.1.*
Substep 0.2: $P(\alpha)+P_\#(\alpha) \neq \emptyset$; hence determine if α contains more labelled points with a given label than γ. For each j, $1 \leqslant j \leqslant N_\alpha$, do:
Compare $A_j(\alpha)$ with labels $A_1(\gamma)$, ..., $A_{N_\gamma}(\gamma)$ for a label, say $A_k(\gamma)$, such that $A_j(\alpha) = A_k(\gamma)$. If no such label $A_k(\gamma)$ exists, go to *substep 4.2* (in this case, the shape rule cannot possibly apply to γ and $T_{R,\gamma} = \emptyset$), otherwise, if $n_j(\alpha) > n_k(\alpha)$, go to *substep 4.2* (in this case, again the shape rule cannot possibly apply to α and $T_{R,\gamma} = \emptyset$). [Notice that if $P(\gamma)+P_\#(\gamma)$ is organized as a balanced tree (see Krishnamurti, 1980), this step requires at worst $O(N_\alpha \mathrm{lb} N_\gamma)$ time.]

Step 1: The construction of the fixed distinguishable points in DP_α.
Substep 1.1: Let *dptr* be the pointer to the last entry in DP_α; set $dptr \leftarrow k \leftarrow 0$.
Substep 1.2: $k \leftarrow k+1$. Let the kth element in $\Pi(\gamma)$ be $\pi(k)$.
Substep 1.3: Compare $A_{\pi(k)}$ with labels $A_1(\alpha)$, ..., $A_{N_\alpha}(\alpha)$ for a label, say $A_j(\alpha)$, such that $A_{\pi(k)}(\gamma) = A_j(\alpha)$. If no such label $A_j(\alpha)$ exists, go to *substep 1.2.*
Substep 1.4: $A_{\pi(k)}(\gamma) = A_j(\alpha)$ and from *substep 0.2* $n_{\pi(k)}(\gamma) \geqslant n_j(\alpha)$; set $n \leftarrow n_{\pi(k)}(\gamma)$, $f' \leftarrow dptr$. For $1 \leqslant i \leqslant n_j(\alpha)$ and either until all the points in $P_j(\alpha)$ have been examined or until $dptr = f$ do the following:

Let p_i be the ith point in $P_j(\alpha)$, and compare p_i with the points in $DP_\alpha(1, ..., f')$ for distinctness. [Since all the points in $P_j(\alpha)$ are distinct, p_i need not be compared with the points, if any, in $DP_\alpha\ (> f')$.]
If p_i is distinct from the points in DP_α, set $dptr \leftarrow dptr + 1$, $DP_\alpha(dptr) \leftarrow p_i$, $map(dptr) \leftarrow \pi(k)$ (where *map* contains the index of the set of labelled points in γ which have the same label as p_i), and $n \leftarrow n+1$.
If $dptr = f$ and $type = 1$ then, if $n > 2$, set $dptr \leftarrow f \leftarrow 2$ (that is, shape rules of type 1 can be treated as shape rules of type 2), otherwise (for $n \leqslant 2$) compare the slopes of the lines formed by $\{DP_\alpha(1), DP_\alpha(2)\}$ and $\{DP_\alpha(2), DP_\alpha(3)\}$ for equality. If equality holds (the lines are colinear and hence do not form a triangle) set $dptr \leftarrow 2$ and $n \leftarrow n+1$. [In this case, the last point added to DP_α is eliminated. This can be done since colinearity of lines is transitive which ensures that for shape rules of type 1, irrespective of the choices for the first two points in DP_α there will always be a third point which forms a triangle with these two points. Since *substeps 0.2* and *1.3* ensure that all the labelled points in $P(\alpha) + P_\#(\alpha)$ will be examined, this third point will always be found. Of course, it is assumed that the shape rule is properly type classified.]
Substep 1.5: All the points in $P_j(\alpha)$ have been examined or else $dptr = f$. If $dptr < f$, go to *substep 1.2*, otherwise (that is, $dptr = f$) go to *substep 2.1* except in the case of $type = 3$, that is, set $DP_\alpha(2) \leftarrow$ a single point in $P_1(\alpha)$ and then go to *substep 2.1*.

Step 2: The generation of all possible mappings from $DP_\alpha(j)$ to $P_{map(j)}(\gamma)$, $1 \leqslant j \leqslant f$, and the mappings for shape rules of types 1, 2, 3, and 4.
Each mapping may be viewed as a combination of f distinct points one from each of the f sets $P_{map(1)}(\gamma)$, ..., $P_{map(f)}(\gamma)$. However, DP_α may contain points which correspond to labelled points with the same label in which case their corresponding points of γ are selected from the same labelled point set. That is, their 'map' values are the same. Furthermore γ may contain two different labelled points which share the same coordinates. In order to ensure that the f points are distinct, a variable *mark* is associated with each labelled point of γ, which essentially takes on two values: that is, +1 if p is available for selection, and −1 if p is used in some pair $\langle DP_\alpha(j), p\rangle$, for $1 \leqslant j \leqslant f$.

For each point p in $DP_\alpha(1, ..., f)$ the corresponding point p' from $P_{map(1, ..., f)}(\gamma)$ is stored in $DP_\gamma(1, ..., f)$. That is, for $1 \leqslant j \leqslant f$, $\langle DP_\alpha(j), DP_\gamma(j)\rangle$ describes a point-point mapping. The combinations—that is, the points in DP_γ—are generated via the standard technique of backtrack programming (see, for instance, Krishnamurti and Roe, 1979, pages 198–201). The following is a step by step translation of figure 13 which gives a pictorial illustration of backtrack programming. The dotted lines indicate the order in which the combinations are generated.

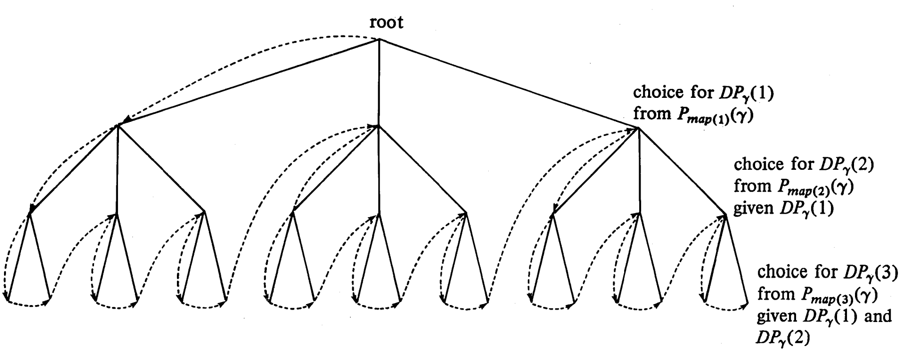


Figure 13. Search tree for generating the point-point mappings between DP_α and DP_γ.

Substep 2.1: *dptr* points to the current entry in DP_γ. Set $dptr \leftarrow 0$.
Substep 2.2: Set $dptr \leftarrow dptr+1$, and $j \leftarrow 1$.
Substep 2.3: If $j > n_{map(dptr)}$ go to *substep 2.5*, otherwise [for $j \leqslant n_{map(dptr)}$; that is, there are still some unexamined points in $P_{map(dptr)}(\gamma)$] let p'_j be the jth point in $P_{map(dptr)}(\gamma)$.
Substep 2.4: If $mark(p'_j) < 0$ (p'_j has already been selected in DP_γ) set $j \leftarrow j+1$ and go to *substep 2.3*; otherwise, compare p'_j with the points, if any, in DP_γ. If p'_j shares the same coordinates with any other point in DP_γ, set $j \leftarrow j+1$ and go to *substep 2.3*; if not (p'_j has not been selected and is distinct from the points, if any, in DP_γ) do the following:
Set $DP_\gamma(dptr) \leftarrow p'_j$ and $index(dptr) \leftarrow j$ [*index* points to the indices of the points in $P_{map(1,\ldots,dptr)}(\gamma)$ that have been selected in DP_γ]. If $dptr < f$, set $mark(p'_j) \leftarrow -mark(p'_j)$ (notice that this assignment marks p'_j as selected) and go to *substep 2.2*; otherwise ($dptr = f$ and DP_γ has been constructed) set $j \leftarrow j+1$ and go to *substep 2.3* after performing one of the following group of statements depending upon the value of f:
(a) for $f = 3$ (that is, for shape rules of type 1) determine if the points in DP_γ form a triangle, and if so, determine, if the triangles formed by the points in DP_α and DP_γ are similar. Again, if so, determine the coefficients of the transformation τ, defined by the mapping $DP_\alpha(1, \ldots, 3) \leftrightarrow DP_\gamma(1, \ldots, 3)$, by means of equations (10) through (12). If $\tau(\alpha) \leq \gamma$, set $nt \leftarrow nt+1$ and push τ into $T_{R,\gamma}$.
(b) for $f = 2$ (that is, for shape rules of type 2) construct the similar triangles defined with respect to the mapping $DP_\alpha(1, \ldots, 2) \leftrightarrow DP_\gamma(1, \ldots, 2)$ via the construction given by equations (14) through (16), and determine the coefficients of the two transformations τ_1^* and τ_2^*, given by equation (17), by means of equations (10) through (12). If either $\tau_1^*(\alpha) \leq \gamma$ or $\tau_2^*(\alpha) \leq \gamma$ then increment nt accordingly and push the appropriate transformation(s) into $T_{R,\gamma}$.
(c) for $f = 1$ (that is, for shape rules of types 3 and 4), if $type = 4$ and $s(\alpha) = s_\phi$, then τ is a translation which takes $DP_\alpha(1)$ onto $DP_\gamma(1)$, and in which case increment τ and push τ into $T_{R,\gamma}$; otherwise, perform the following *substeps*:
Substep 2.4.1: If $type = 3$ go to *substep 2.4.4*; otherwise, set $i \leftarrow 1$.
Substep 2.4.2: Let $n_\cdot(\alpha)$ denote the cardinality of $P_\cdot(\alpha)$. If $i > n_\cdot(\alpha)$, go to *substep 2.4.6*; otherwise, let p_i be the ith point in $P_\cdot(\alpha)$.
Substep 2.4.3: If $p_i = DP_\alpha(1)$, set $i \leftarrow i+1$ and go to *substep 2.4.2*; otherwise, set $DP_\alpha(2) \leftarrow p_i$.
Substep 2.4.4: For each k, $1 \leqslant k \leqslant M_\gamma$ do the following: Let $\langle \mu_k, \nu_k \rangle$ be the line descriptor for the maximal lines in $s_k(\gamma)$. Let p' be the $\perp$-intersection point of $DP_\gamma(1)$ and the line with line descriptor $\langle \mu_k, \nu_k \rangle$. If $p' \neq DP_\gamma(1)$ and $type = 3$ set $DP_\gamma(2) \leftarrow p'$ and determine the transformations τ_1^* and τ_2^*, as in (b) of *substep 2.4*, each of which is examined for membership in $T_{R,\gamma}$.
On the other hand, if $DP_\gamma(1) = p'$ and $type = 4$, then do the following loop: for each end point p'_2 of a maximal line in $s_k(\gamma)$ such that $p'_2 \neq DP_\gamma(1)$ set $DP_\gamma(2) \leftarrow p'_2$ and, as in (b) of *substep 2.4*, determine transformations τ_1^* and τ_2^*, defined by the mapping $DP_\alpha(1, \ldots, 2) \leftrightarrow DP_\gamma(1, \ldots, 2)$, each of which is examined for membership in $T_{R,\gamma}$.
Substep 2.4.5: All the subshapes $s_k(\gamma)$, $1 \leqslant k \leqslant M_\gamma$ have been examined. If $type = 4$, set $i \leftarrow i+1$ and go to *substep 2.4.2*.
Substep 2.4.6: The mappings for shape rules of types 3 and 4 for the given $DP_\alpha(1) \leftrightarrow DP_\gamma(1)$ mapping have been generated.
Substep 2.5: All the points in $P_{map(dptr)}(\gamma)$ have been examined. That is, all the choices for $DP_\gamma(dptr)$ given the choices for $DP_\gamma[1, \ldots, (dptr-1)]$ have been examined. Backtrack and select the next choice, if possible, for $DP_\gamma(dptr-1)$ and continue. Set $dptr \leftarrow dptr-1$; and if $dptr > 0$, set $p'_j \leftarrow DP_\gamma(dptr)$, $mark(p'_j) \leftarrow -mark(p'_j)$ (note

that this assignment unmarks p_j'), and $j \leftarrow index(dptr)+1$ and then go to *substep 2.3*; otherwise ($dptr = 0$ and all possible mappings between the fixed labelled points of α and the corresponding labelled points of γ have been considered) go to *substep 4.1*.
Step 3: The mappings for shape rules of type 5.
Substep 3.1: If $M_\alpha > 1$, go to *substep 3.4*.
Substep 3.2: Otherwise $M_\alpha = 1$, that is, all the maximal lines in $s(\alpha)$ are colinear. By proviso 5, each transformation τ for which $\tau(\alpha) \lesssim \gamma$ must map a pair of end points of maximal lines in $s(\alpha)$ to corresponding end points of maximal lines in $s(\gamma)$. Let $\epsilon(p)$ be the function which takes on the value t in the case that p is the tail of a maximal line and the value h for p otherwise. Let $t = -h$, and let $\{p_1, p_2\}$ and $\{p_1', p_2'\}$ denote the corresponding pairs of end points where $p_1 < p_2$ and $p_1' < p_2'$. Then, one or both of the following correspondences:

$$\langle p_1, p_2\rangle \leftrightarrow \langle p_1', p_2'\rangle\,, \qquad \langle p_1, p_2\rangle \leftrightarrow \langle p_2', p_1'\rangle\,, \tag{18}$$

may yield transformations τ such that $\tau(\alpha) \lesssim \gamma$. For each pair of end points of maximal lines in $s(\alpha)$, $\langle p_1, p_2\rangle$, $p_1 < p_2$, carry out the following:
If $\{p_1, p_2\}$ is a maximal line is $s(\alpha)$, perform *substep 3.2.1* for each k, $1 \leqslant k \leqslant M_\gamma$.
Substep 3.2.1: For each maximal line l, $l = \langle p_1', p_2'\rangle$ in $s_k(\gamma)$ the following is determined:
For each correspondence in expressions (18) the two transformations τ_1^* and τ_2^* given by conditions (17) via the construction provided by equations (14) through (16), and by means of equations (10) through (12) to compute their coefficients determine if $\tau_1^*(\alpha) \lesssim \gamma$. If so [then τ_2^* also satisfies $\tau_2^*(\alpha) \lesssim \gamma$] push both transformations into $T_{R,\gamma}$ and increment nt by 2. Notice that in this case for each k, $1 \leqslant k \leqslant M_\gamma$, only $2m_k(\gamma)$ transformations have to be tested for membership in $T_{R,\gamma}$; the remaining $2m_k(\gamma)$ transformations are either automatically accepted or rejected as members of $T_{R,\gamma}$.

Otherwise, $\{p_1, p_2\}$ is not a maximal line; set $e_1 \leftarrow \epsilon(p_1)$, $e_2 \leftarrow \epsilon(p_2)$ and perform *substep 3.2.2*.
Substep 3.2.2: For each pair of end points of maximal lines in $s_k(\gamma)$, $\langle p_1', p_2'\rangle$, $p_1' < p_2'$ do the following:
Set $e_1' \leftarrow \epsilon(p_1')$ and $e_2' \leftarrow \epsilon(p_2')$. If $2e_1+e_2 = 2e_1'+e_2'$, compute and test transformations τ_1^* and τ_2^* defined by the first of correspondences (18). As before, only one of these transformations, say τ_1^*, need be examined for membership in $T_{R,\gamma}$. The other is automatically accepted or rejected for membership in $T_{R,\gamma}$. If, in addition, $e_1 = -e_2$, compute and test transformations τ_1^* and τ_2^* defined by the second of correspondences (18) in the above manner. On the other hand, if $2e_1+e_2 = -(2e_1'+e_2')$, compute and test transformations τ_1^* and τ_2^* defined by the second of correspondences (18), again, in the above manner.

Notice that in this substep for each k, $1 \leqslant k \leqslant M_\gamma$, again only half the possible $(1+a)m_k(\gamma)[m_k(\gamma)-1+a]$ transformations have to be tested for membership in $T_{R,\gamma}$, where $a = |e_1-e_2|$.
Substep 3.3: All the mappings for shape rules of type 5 with $M_\alpha = 1$ have been considered, and so *step 3* is exited by going to *substep 4.1*.
Substep 3.4: With $M_\alpha > 1$, determine if $s(\gamma)$ contains at least M_α maximal lines all parallel to one another. Recall that for j, $1 \leqslant k \leqslant M_\gamma$, $\mu_j(\gamma)$ is the slope of the jth subshape $s_j(\gamma)$. Since the subshapes of $s(\gamma)$ are arranged according to increasing line descriptor values, it follows that for any $j \neq k$, $1 \leqslant j < k \leqslant M_\gamma$, if $\mu_j(\gamma) = \mu_k(\gamma)$, then $\mu_i(\gamma) = \mu_j(\gamma)$, for all $j < i \leqslant k$. Set $\gamma ptr \leftarrow k \leftarrow 0$.
Substep 3.5: Set $k \leftarrow k+1$.
Substep 3.6: If $k > M_\gamma - M_\alpha + 1$, go to *substep 3.7*. If $\mu_k(\gamma) \neq \mu_{k+M_\alpha-1}(\gamma)$, go to *substep 3.5*; otherwise, determine the largest j, $k+M_\alpha-1 \leqslant j \leqslant M_\gamma$, such that

$\mu_j(\gamma) = \mu_k(\gamma)$ and, if $j < M_\gamma$, $\mu_{j+1}(\gamma) \neq \mu_k(\gamma)$. Then, set $\gamma ptr \leftarrow \gamma ptr+1$, $\gamma stack(\gamma ptr) \leftarrow \langle k, j \rangle$, and $k \leftarrow j$, and go to *substep 3.5*.
Substeps 3.5 and *3.6* require $O(M_\gamma)$ time to construct $\gamma stack$ which contains the indices of the subshapes with the least and greatest line descriptor values from a collection of parallel subshapes.
Substep 3.7: In this substep, γptr contains the number of entries in $\gamma stack$. If $\gamma ptr = 0$, that is, $\gamma stack$ is empty and the shape rule cannot possibly apply, go to *substep 4.2*, otherwise perform *substep 3.8*.
Substep 3.8: For each j, $1 \leqslant j \leqslant M_\alpha$, let $\check{p}$ be the least tail of the maximal lines in $s_j(\alpha)$, $\hat{p}$ the greatest head of the maximal lines in $s_j(\alpha)$, and $\mathrm{H}^*(\check{p})$ the H-intersection point of $s_j(\alpha)$ and $s_{j'}(\alpha)$, $j \neq j'$, with respect to $\check{p}$. Then, for each i, $1 \leqslant i \leqslant \gamma ptr$, set $\langle bot, top \rangle \leftarrow \gamma stack(i)$, and then, for each k, $bot \leqslant k \leqslant top$, let $\check{p}'$ be the least tail of the maximal lines in $s_k(\gamma)$, and let $\hat{p}'$ be the greatest head of the maximal lines in $s_k(\gamma)$. Then, for each $k' \neq k$, $bot \leqslant k' \leqslant top$, perform the following: Let $\mathrm{H}_k(p')$ be the H-intersection point of $s_k(\gamma)$ and $s_{k'}(\gamma)$ with respect to $\check{p}'$, compute the transformations τ_1^* and τ_2^* defined by the mapping $\langle \check{p}, \mathrm{H}^*(\check{p}) \rangle \leftrightarrow \langle \check{p}', \mathrm{H}_k(\check{p}') \rangle$, and determine for each j'', $j'' \neq j$, $j'' \neq j'$, $1 \leqslant j'' \leqslant M_\alpha$, whether there is a k'', $bot \leqslant k'' \leqslant top$, $k'' \neq k$, $k'' \neq k'$, such that $\tau_1^*[\psi_{j''}(\alpha)] = \psi_{k''}(\gamma)$. Notice that for all j'', $1 \leqslant j'' \leqslant M_\alpha$, $\tau_2^*[\psi_{j''}(\alpha)] = \tau_1^*[\psi_{j''}(\alpha)]$; and it may be supposed, without loss in generality, that τ_1^* preserves in $\tau_1^*(\alpha)$ the original ordering on the maximal lines in α. If so, *and* $\hat{p} \nless \tau_1^*(\hat{p})$, then the following are determined and tested. For each of the end points p_1 of the maximal lines in $s_j(\alpha)$, let τ' be the translation from $\tau_1^*(p_1)$ to $\check{p}'$, and let τ'' be the translation from $\tau_2^*(p_1)$ to $\check{p}'$. Clearly the transformations defined by the mapping $\langle p_1, \mathrm{H}^*(p_1) \rangle \leftrightarrow \langle \check{p}', \mathrm{H}_{k'}(\check{p}') \rangle$ are given by $\tau_1 \equiv \tau'(\tau_1^*)$ and $\tau_2 \equiv \tau''(\tau_2^*)$. Then, for each end point p_1' of maximal lines in $s(\gamma)$, let τ be the translation from $\check{p}'$ to p_1'. Clearly, the transformations defined by the mapping $\langle p_1, \mathrm{H}^*(p_1) \rangle \leftrightarrow \langle p_1', \mathrm{H}_{k'}(p_1') \rangle$ are given by $\tau_1' \equiv \tau(\tau_1)$ and $\tau_2' \equiv \tau(\tau_2)$. Now let $\epsilon(p)$ be the function which takes the value t if p is the tail of a maximal line and the value h for p otherwise, and determine whether $\tau_1'(\alpha) \leq \gamma$, if $\epsilon(p_1) \neq \epsilon(p_1')$ *and* $\hat{p}' \nless \tau_1'(\hat{p})$ *and* $\tau_1'(\check{p}) \nless \check{p}'$ or $\tau_2'(\alpha) \leq \gamma$, if $\epsilon(p_1) \neq \epsilon(p_1')$ *and* $\tau_2'(\hat{p}) \nless \check{p}'$ *and* $\hat{p}' \nless \tau_2'(\check{p})$. If so, add the appropriate transformation and increment nt. Figure 14 illustrates how the transformations τ_1, τ_2, τ_1', and τ_2' are determined given the transformations τ_1^* and τ_2^*.
Notice that in this substep, for each $\check{p}$, $\mathrm{H}^*(\check{p})$ is kept fixed.
Substep 3.9: All the mappings for shape rules of type 5 with $M_\alpha \leqslant 1$ have been considered, and so *step 3* is exited by going to *substep 4.1*.

Step 4: Finishing touches
Substep 4.1: Does the shape rule apply? If $nt \neq 0$, go to *substep 4.3*, otherwise ($T_{R,\gamma}$ is empty) go to *substep 4.2*.
Substep 4.2: "Shape rule does not apply". Exit algorithm.
Substep 4.3: "Shape rule applies".
$\tau_1, \tau_2, ..., \tau_{nt}$ are the transformations under which R applies to γ. Exit algorithm. Note that for shape rules of types 4 and 5 some of these transformations may be duplicated. However, for this algorithm this is immaterial.
Remark: The designer may choose one of the transformations in $T_{R,\gamma}[T_{R,\gamma} \neq \emptyset]$, say τ_j, $1 \leqslant j \leqslant nt$, and apply the shape rule $\gamma \leftarrow [\gamma - \tau_j(\alpha - \beta)] + \tau_j(\beta - \alpha)$. For keeping track of rule application a *rule stack* may be employed. Each time rule R is applied under transformation τ_j $\langle R, \tau_j, \beta^* \rangle$ is pushed into a 'rule stack', where β^* is the shape given by the shape intersection $\beta^* \leftarrow \gamma \cdot \tau_j(\beta - \alpha)$.
Notice that β^* is computed before γ is modified. Thus, if the designer should wish to return to an earlier state in the shape generation this can be done computationally by means of the backtracking identity $\gamma \leftarrow [\gamma - \tau_j(\beta - \alpha)] + \tau_j(\alpha - \beta) + \beta^*$.

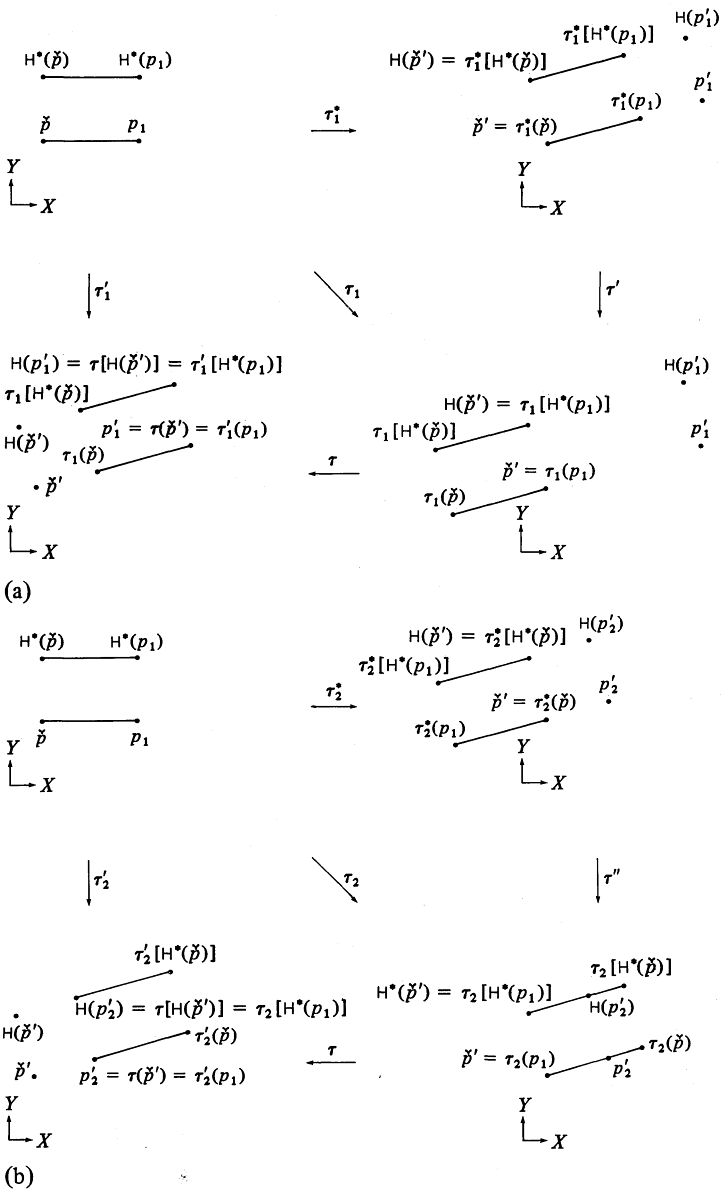


Figure 14. Obtaining the transformations τ_1' and τ_2' from τ_1^* and τ_2^* via a sequence of translations.

Data structures

In this section the relevant data structures necessary for an efficient implementation of the shape rule application algorithm are described. An Algol-like translation of the algorithm which incorporates the data structures is presented.

The data structures necessary to represent labelled shapes have been discussed in Krishnamurti (1980). Each labelled shape is represented by either a pair of balanced binary trees or by a pair of linked lists. For each type of representation, one member of the pair houses the maximal lines while the other houses the labelled points. However, as has just been seen, for shape rule application, it is convenient to represent the current labelled shape γ by the ordered triple $\langle s, P^*, \Pi \rangle$ where P^* is the set union $P + P_{\#}$, and a shape rule R by the ordered four-tuple $\langle \alpha, \alpha - \beta, \beta - \alpha, type \rangle$.

In the remainder of this section attention will be devoted to the data structures for γ and R, and in particular, the changes and additions made to the original data structures will be emphasized. It should be noted that the changes will not detract from the effectiveness of the data structures in implementing the shape algorithms described in Krishnamurti (1980).

The current labelled shape γ
Consider the data structures for the maximal lines. The maximal lines are housed in a balanced binary tree. Each tree node contains a field, N, which is a pointer to a linked list which represents a list of multiple colinear maximal lines. That is, every maximal line in the list has the same line descriptor which is stored in the *key* field of the tree node. The linked list consists of nodes of which there are two kinds. The first is a 'header' node. The second kind is a 'line' node, which as the name implies represents a maximal line. Both kinds of nodes have three fields described below:

line node

head	*tail*	*next*

,

tail and *head* are pointers to an array which stores the coordinates of the points, and *next* is a pointer to next node in the list. The list is ordered in the following sense: for any *node*, coordinates of *tail*[*node*] < coordinates of *head*[*node*] and coordinates of *head*[*node*] < coordinates of *tail*[*next*[*node*]].

header node

thread	*bot*	*top*

,

top points to the first 'line' node in the list, *bot* points to the last 'line' note in the list, and *thread* is utilized in the algorithm for constructing a circular list which contains all the tree nodes representing maximal lines which share the same slope. In fact, the 'header' node is useful only for shape rules of type 5. For any *tree node*, *N*[*tree node*] points to a 'header' node of a list of colinear maximal lines.

Now, consider the data structure for the set of labelled points. The labelled points are housed in a balanced binary tree. This balanced tree also incorporates the points of intersection in the following way. Each tree node contains a field, N, which points to a linked list which represents all the points which have the same label which is stored in the *key* field of the tree node. The 'key' takes on one of the following values:

$$key = \begin{cases} \text{a label } A, & \text{if node represents all labelled points of the form } p{:}A, \\ \#, & \text{if node represents the points of intersection.} \end{cases}$$

The ordering on the labels is assumed to be $A_1 < A_2 < \ldots < \#$.

The linked list consists of nodes each of which represents a point. As in the case of maximal lines, there are two kinds of nodes: 'header' and 'point' nodes. Each node has three fields. The form for 'point' nodes is given by

point node

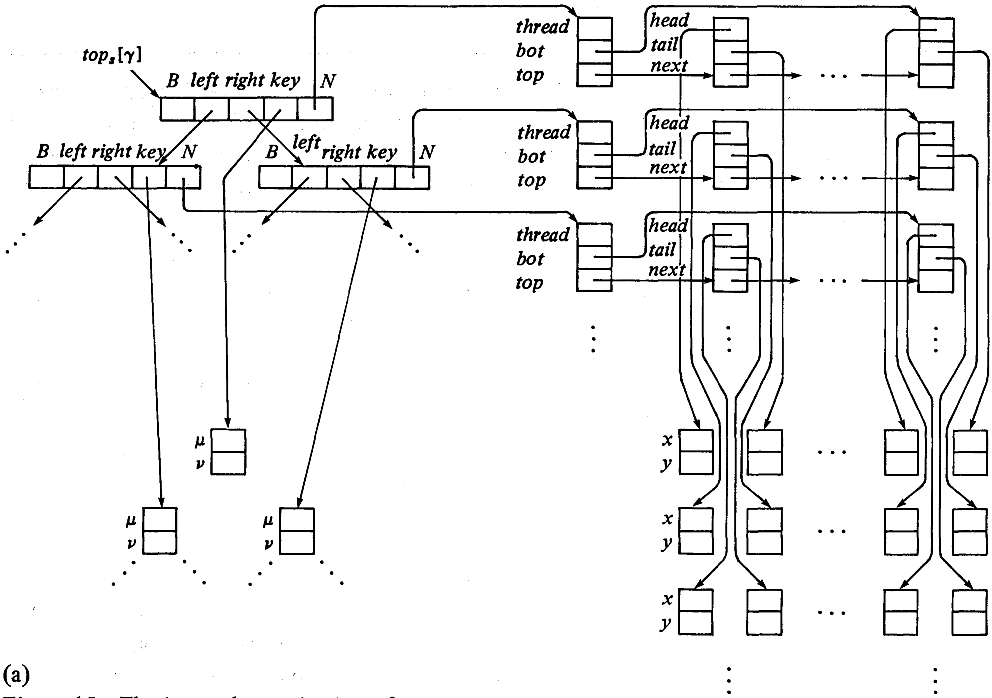
point	*mark*	*next*

,

point is a pointer to an array which stores the coordinates of the point represented by the node, *next* points to the next node in the list, and *mark* is a positive integer which takes on values:

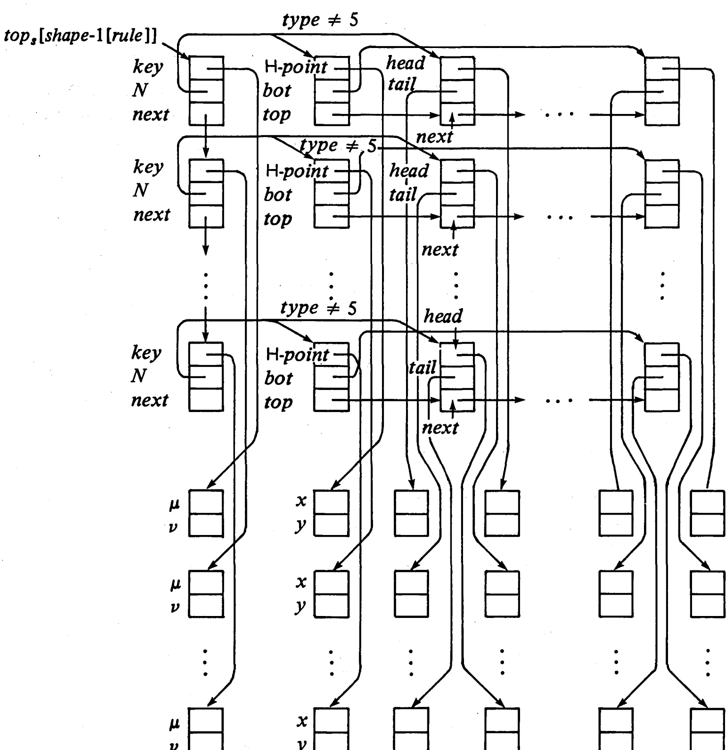
$$mark = \begin{cases} \text{number of pairs of maximal lines which} & \\ \text{meet at the point represented by the list node,} & \text{if } key = \#, \\ +1, & \text{otherwise.} \end{cases}$$

mark is mainly employed as the 'mark' variable for the shape rule application algorithm (see *step 2*), but it also serves to indicate the presence of a point in the



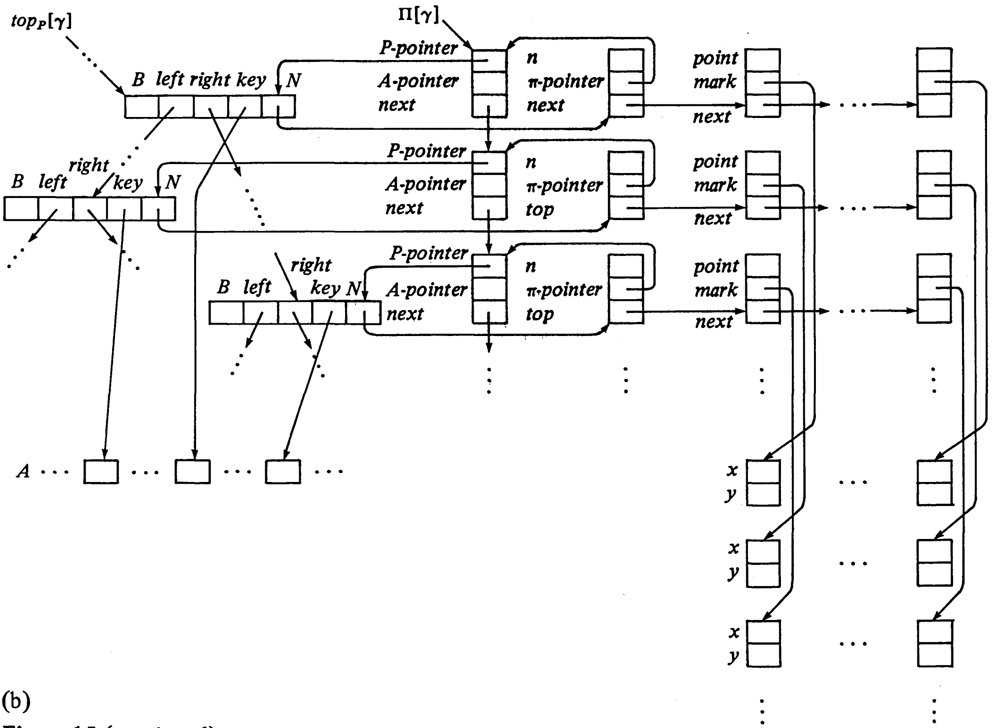
(a)

Figure 15. The internal organization of γ.

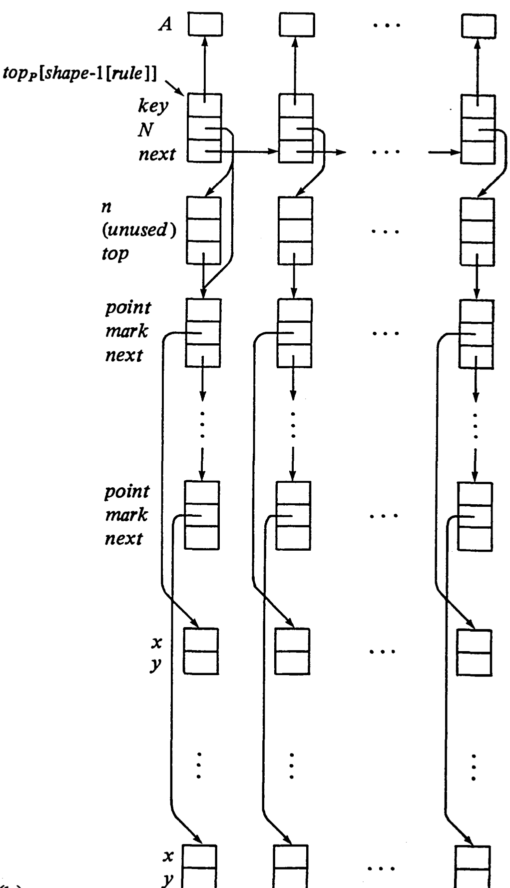


(a)

Figure 16. The internal organization of α.



(b)
Figure 15 (continued)



(b)
Figure 16 (continued)

shape in the following sense: every time a *new instance* of the point occurs in the shape *mark* is updated. That is, *mark* is updated, in the case of points of intersection whenever a node is added to or deleted from the data structure for maximal lines during a shape union or shape difference. In this way, the points of intersection do not have to be recomputed each time a shape rule is applied. Instead, *mark* can be updated after each execution of a shape union or a shape difference; all that has to be done is to keep track of all the data nodes added to or deleted from the data structure for the maximal lines, and the rest is straightforward.

header node

n	*π-pointer*	*top*

,

n is the number of nodes (hence, the number of points) in the list, *top* is a pointer to the first node in the list, and *π-pointer* is a pointer to a node in a list which maintains the order of the point set cardinalities and is described in detail below.

Π, the permutation of the indices of the point sets in $P+P_{\#}$ is represented by a linked list. Each node consists of three fields:

π-node

P-pointer	*A-pointer*	*next*

,

next is a pointer to the succeeding node in the list. *P-pointer* is a pointer to a node in the balanced tree for the labelled points. For any *tree node*, *N*[*tree node*] points to the header node of the list of labelled points having the same label. Then, *P-pointer* satisfies the following identity:

P-pointer[*π-pointer*[*N*[*tree node*]]] = *tree node*.

A pictorial description of the data structures representing the current labelled shape γ is given in figure 15. Recall from Krishnamurti (1980) that the tree nodes belong to the data class *BTREE* and list nodes belong to the data class *LIST*.

Algorithm SHAPE RULE APPLICATION (R, γ)
¶ This procedure constructs the set $T_{R,\gamma}$ ¶
¶ Step 0: Initialization. *nt* is the number of transformations, and *error* is a Boolean flag which is **true** only when it is determined that R does not apply to γ ¶
$\alpha \leftarrow$ *shape-1*[R]
$f \leftarrow$ (**if** *type*[R] $\leqslant 3$ **then** $4 -$ *type*[R] **else** $5 -$ *type*[R])
nt $\leftarrow 0$
error $\leftarrow$ **false**
¶ For each label A_α determine whether it is also the label for some labelled point(s) in γ. If so, does the number of points in α having label A_α exceed the number of points in γ having label A_α? Note that for shape rules of types 3 and 4 with $top_s[\alpha] \neq$ **null**, $top_P[\alpha]$ points to a node which represents either a single ⊥-intersection point or a list of end points of maximal lines none of which share the same coordinates as the distinct labelled point in α ¶
P_α-*node* $\leftarrow$ (**if** ($f = 1$) **and** ($top_s[\alpha] \neq$ **null**) **then** *next* [$top_P[\alpha]$] **else** $top_P[\alpha]$)
while (P_α-*node* $\neq$ **null**) **and** (**not** *error*)
do {
  $A_\alpha \leftarrow A[key[P_\alpha\text{-}node]]$
  P_γ-*node* $\leftarrow top_P[\gamma]$
  ¶ Search balanced tree rooted at $top_P[\gamma]$ for a node whose 'key' equals A_α ¶
  while (P_γ-*node* $\neq$ **null**) **and** ($A_\alpha \neq (A_\gamma \leftarrow A[key[P_\gamma\text{-}node]])$)
  do P_γ-*node* $\leftarrow$ (**if** $A_\alpha < A_\gamma$ **then** *left*[P_γ-*node*] **else** *right*[P_γ-*node*])
  if (P_γ-*node* $\neq$ **null**) **and** ($n[N[P_\alpha\text{-}node]] \leqslant n[N[P_\gamma\text{-}node]]$)
  then { *A-pointer*[*π-pointer*[$N[P_\gamma\text{-}node]$]] $\leftarrow P_\alpha$-*node*
    P_α-*node* $\leftarrow$ *next*[P_α-*node*] }
  else *error* $\leftarrow$ **true** }
¶ For shape rules of types 3, 4, and 5, place in *γstack* the nodes, in order, from the balanced tree rooted at $top_s[\gamma]$ ¶
if (**not** *error*) **and** ($f \leqslant 1$) **and** ((L_α-*node* $\leftarrow top_s[\alpha]$) $\neq$ **null**)
then {
  γptr $\leftarrow j \leftarrow 0$
  stack[*sptr* $\leftarrow 1$] $\leftarrow L_\gamma$-*node* $\leftarrow top_s[\gamma]$ ¶ *stack* is a temporary array ¶
  while *sptr* > 0
  do {
    if $j = 0$ **then while** *left*[L_γ-*node*] $\neq$ **null do** { *stack*[*sptr* $+\leftarrow 1$] $\leftarrow L_\gamma$-*node*
      L_γ-*node* $\leftarrow$ *left*[L_γ-*node*] }
    γstack[*γptr* $+\leftarrow 1$] $\leftarrow L_\gamma$-*node*
    $\langle L_\gamma\text{-}node, j\rangle \leftarrow$ **if** *right*[L_γ-*node*] $\neq$ **null then** $\langle$*right*[L_γ-*node*], 0$\rangle$ **else** $\langle$*stack*[(*sptr* $- \leftarrow 1) + 1$], 1$\rangle$ } }

Algorithm 1. (This algorithm continues until page 38.)

The shape rule R

Each shape rule is represented by a node in a data class, say *RULE*, each of whose nodes consists of four fields:

rule node

shape-1	*shape*-2	*shape*-3	*type*

,

shape-i ($i = 1, 2, 3$) is a pointer to pairs of roots of data structures each pair representing a labelled shape. That is,
shape-1: represents the labelled shape α, $\alpha = \langle s(\alpha), P^*(\alpha)\rangle$
shape-2: represents the labelled shape $\alpha - \beta$, $\alpha - \beta = \langle s(\alpha - \beta), P(\alpha - \beta)\rangle$
shape-3: represents the labelled shape $\beta - \alpha$, $\beta - \alpha = \langle s(\beta - \alpha), P(\beta - \alpha)\rangle$.
The shapes may conveniently be stored in a data class, say *SHAPE*, each of whose nodes consist of two fields:

shape node

top_S	top_P

,

top_S is a pointer to the 'top node' in the linked list which represents the maximal lines, and top_P is a pointer to the 'top node' in the linked list which represents the

```
¶ Step 1: The determination of the f labelled points in DPα. The following loop is performed until the fixed points
  have been determined. Provided that the rule type has been properly assigned, this loop will halt since from step 0
  it has been determined that there are a sufficient number of labelled points in γ. Note that this loop is executed
  only if error is false and f > 0 ¶
dptr ← (if not error then 0 else f)
π-node ← Π[γ]   ¶ π-node is set initially to the first node in the list representing the permutation Π(γ) ¶
while dptr < f
do { if A-pointer[π-node] ≠ null
     then { f' ← dptr
            αpoint ← next[N[A-pointer[π-node]]]
            γheader ← N[P-pointer[π-node]]
            nγ ← n[γheader]
            A-pointer[π-node] ← null
            while (αpoint ≠ null) and (dptr < f)
            do { p ← point[αpoint]
                 for j ← 1, j while (j ≤ f') and (⟨x, y⟩[p] ≠ ⟨x, y⟩[DPα[j]]) do j +← 1
                 if j > f'
                 then { ¶ p is distinct from the points, if any, in DPα ¶
                        DPα[dptr +← 1] ← p
                        map[dptr] ← γheader
                        nγ −← 1
                        if dptr = f
                        then { if f = 3
                               then { if nγ ≥ 2
                                      then { ¶ Treat shape rules of type 1 as shape rules of type 2 ¶
                                             f ← dptr ← 2 }
                                      else { ¶ Let ⟨xj, yj⟩ denote the coordinates of DPα[j]. That is, for
                                               1 ≤ j ≤ 3, ⟨xj, yj⟩ ← ⟨x, y⟩[DPα[j]] ¶
                                             μ12 ←p ((y1 − y2)/(x1 − x2))
                                             μ23 ←p ((y2 − y3)/(x2 − x3))
                                             ¶ The points in DPα form a triangle only if the slopes μ12 and μ23 are
                                               unequal ¶
                                             if μ12 = μ23
                                             then { dptr ← 2
                                                    nγ +← 1 }
                                             else { l12 ←p ((y1 − y2)² + (x1 − x2)²)
                                                    l23 ←p ((y2 − y3)² + (x2 − x3)²)
                                                    l31 ←p ((y3 − y1)² + (x3 − x1)²) } } } }
                 αpoint ← next[αpoint] } }
     π-node ← next[π-node] }
¶ For shape rules of type 3 set DPα[2] to a ⊥-intersection point of DPα[1] ¶
if (not error) and (type[R] = 3) then DPα[2] ← point[N[topP[α]]]
¶ Reset the A-pointer values to null ¶
if f > 0 then while π-node ≠ null do { A-pointer[π-node] ← null
                                        π-node ← next[π-node] }
```

Algorithm 1 (continued)

```
¶ Step 2: Generate the mappings for shape rules of types 1, 2, 3, and 4. This step takes the form of an iterative
  version of backtrack programming. Note that this step is executed only if error is false and f > 0 ¶
if (not error) and (f > 0) then γpoint ← next[map[dptr ← 1]] else dptr ← 0
while dptr > 0
do  while γpoint ≠ null
    do  if mark[γpoint] > 0
        then  p' ← point[γpoint]
              for j ← 1, j while (j < dptr) and (⟨x, y⟩[p'] ≠ ⟨x, y⟩[DP_γ[j]]) do j + ← 1
              if j = dptr
              then  ¶ p' is distinct from the points, if any, in DP_γ ¶
                    DP_γ[dptr] ← p'
                    if dptr < f
                    then  index[dptr] ← γpoint
                          mark[γpoint] ← −mark[γpoint]
                          γpoint ← map[dptr + ← 1]
                    else  case f in
                            (= 3)  ¶ Shape rules of type 1. Let for j, 1 ≤ j ≤ 3, ⟨x'_j, y'_j⟩ denote the
                                     coordinates of DP_γ[j]. That is ⟨x'_j, y'_j⟩ ← ⟨x, y⟩[DP_γ[j]] ¶
                                   μ'_12 ←_p ((y'_1 − y'_2)/(x'_1 − x'_2))
                                   μ'_23 ←_p ((y'_2 − y'_3)/(x'_2 − x'_3))
                                   ¶ The points in DP_γ form a triangle only if slopes μ'_12 and μ'_23 are
                                     unequal ¶
                                   if μ'_12 ≠ μ'_23
                                   then  for ⟨i, k⟩ ∈ {⟨1, 2⟩, ⟨2, 3⟩, ⟨3, 1⟩}
                                         do  l'_ik ←_p ((y'_i − y'_k)² + (x'_i − x'_k)²)
                                             c_i ←_p (l'_ik/l_ik)
                                         ¶ The corresponding triangles are similar only if c_1 = c_2 = c_3 ¶
                                         if (c_1 = c_2) and (c_2 = c_3) then TRANSFORMATIONS(3,1)
                            (= 2)  ¶ Shape rules of type 2 ¶
                                   TRANSFORMATIONS(2,1)
                            (= 1)  ¶ Shape rules of types 3 and 4 ¶
                                   if L_α-node = null
                                   then  ¶ α has no maximal lines so τ is a translation ¶
                                         TRANSFORMATIONS(1,1)
                                   else  ⟨x'_1, y'_1⟩ ← ⟨x, y⟩[DP_γ[1]]
                                         for j ∈ ⟨1, ..., γptr⟩
                                         do  ¶ Let p̂ be reserved to contain the ⊥-intersection point of
                                               DP_γ[1] and the maximal lines in the list pointed to by
                                               N[γstack[j]] ¶
                                             ⟨μ*, ν*⟩ ← ⟨μ, ν⟩[key[γstack[j]]]
                                             case μ* in
                                               (= 0)         ⟨x, y⟩[p̂] ← ⟨x'_1, ν*⟩
                                               (= ∞)         ⟨x, y⟩[p̂] ← ⟨ν*, y'_1⟩
                                               (otherwise)   ν̂ ←_p (y'_1 + x'_1/μ*)
                                                             x[p̂] ←_p ((ν̂ − ν*)/(μ* + 1/μ*))
                                                             y[p̂] ←_p (μ*x[p̂] + ν*)
                                             equal ← ⟨x, y⟩[p̂] = ⟨x'_1, y'_1⟩
                                             if (not equal) and (type[R] = 3)
                                             then  DP_γ[2] ← p̂
                                                   TRANSFORMATIONS(2,1)
                                             if (equal) and (type[R] = 4)
                                             then  αpoint ← N[top_P[α]]
                                                   while αpoint ≠ null
                                                   do  DP_α[2] ← point[αpoint]
                                                       γline ← top[N[γstack[j]]]
                                                       while γline ≠ null
                                                       do  t' ← tail[γline]
                                                           h' ← head[γline]
                                                           if ⟨x, y⟩[t'] ≠ ⟨x'_1, y'_1⟩
                                                           then  DP_γ[2] ← t'
                                                                 TRANSFORMATIONS(2,1)
                                                           if ⟨x, y⟩[h'] ≠ ⟨x'_1, y'_1⟩
                                                           then  DP_γ[2] ← h'
                                                                 TRANSFORMATIONS(2,1)
                                                           γline ← next [γline]
                                                       αpoint ← next[αpoint]
        γpoint ← next[γpoint]
    ¶ Backtrack ¶
    if (dptr − ← 1) > 0
    then  γpoint ← index[dptr]
          mark[γpoint] ← −mark[γpoint]
          γpoint ← next[γpoint]
```

Algorithm 1 (continued)

```
¶ Step 3: Generate the mappings for shape rules of type 5 ¶
if f > 0 then M_α ← 0 else for M_α ← 0, M_α while L_α-node ≠ null do { M_α +← 1
                                                                     { L_α-node ← next[L_α-node]
if M_α > 1
then
  ¶ α contains at least two parallel maximal lines ¶
  for j1 ← 1, j1 while j1 ≤ (γptr − M_α + 1)
  do
    if (μ* ← μ[key[γstack[j1]]]) = μ[key[γstack[j2 ← j1+M_α−1]]]
    then
      while (j2 < γptr) and (μ* = μ[key[γstack[j2+1]]]) do j2 +← 1
      ¶ Link the nodes in γstack into a circular list ¶
      for k ∈ ⟨j1, ..., (j2 − 1)⟩ do thread [N[γstack[k]]] ← k+1
      thread[j2] ← j1
      ν1 ← ν[key[L_α-node ← top_s[α]]]
      ν2 ← ν[key[next[L_α-node]]]
      while L_α-node ≠ null
      do
        DP_α[1] ← t ← tail[top[αheader ← N[L_α-node]]]
        h ← head[bot[αheader]
        DP_α[2] ← H-point[αheader]
        for k ∈ ⟨j1, j1+1, ..., j2⟩
        do
          DP_γ[1] ← t′ ← tail[top[γheader ← N[γstack[k]]]]
          h′ ← head[bot[γheader]]
          i ← thread[γheader]
          while i ≠ k
          do
            ν* ← ν[key[γstack[i]]]
            ¶ Let p̂ be reserved to contain the H-intersection point of DP_γ[1] ¶
            case μ* in
                (= 0)         ⟨x, y⟩[p̂] ← ⟨x[t′], ν*⟩
                (= ∞)         ⟨x, y⟩[p̂] ← ⟨ν*, y[t′]⟩
                (otherwise)   ν̂ ←_p (y[t′]+x[t′]/μ*)
                              x[p̂] ←_p ((ν̂ − ν*)/(μ* + 1/μ*))
                              y[p̂] ←_p (μ*x[p̂]+ν*)
            DP_γ[2] ← p̂
            TRANSFORMATIONS(2,0)
            ¶ τ1* and τ2* are two possible transformations; rename them, if necessary.
              τ1* preserves the original ordering on the maximal lines in α [see equations (3)
              and (4)]. Let for any τ, τ(ν) be the transformation of the intercept ν [see
              equations (1) and (2)], and τ(p) be the transformation of the point p ¶
            ⟨k′, inc, ≤, j3⟩ ← {if τ1*(ν1) < τ1*(ν2)   then ⟨j1, 1, '>', j2⟩ else ⟨j2, −1, '<', j1⟩}
            ν-match ← τ1*(h) ≤ ⟨x, y⟩[h′]
            L_α-node2 ← top_s[α]
            while L_α-node2 ≠ null
            do
              ν* ← τ1*(ν[key[L_α-node2]])
              while (k′ not ≤ j3) and (ν* ≤ ν[key[γstack[k′]]]) do k′+←inc
              ν-match ← (k′ not ≤ j3) and (ν* = ν[key[γstack[k′]]])
              L_α-node2 ← next[L_α-node2]
            if ν-match
            then
              ¶ τ1* and τ2* maps the line descriptors of the maximal lines in α to
                corresponding line descriptors of maximal lines in γ ¶
              αline ← top[αheader]
              while αline ≠ null
              do
                t1 ← tail[αline]
                h1 ← head[αline]
                Let for j ∈ {1, 2}, p ∈ {t1, h1}, τ_pj be the translation from τj*(p) to t′;
                let τj^p be the composition τ_pj(τj*)
                γline ← top[γheader]
                while γline ≠ null
                do
                  t1′ ← tail[γline]
                  h1′ ← head[γline]
                  Let for p′ ∈ {t1′, h1′}, τ_p′ be the translation from t′ to p′.
                  for ⟨p, p′⟩ ∈ {⟨t1, t1′⟩, ⟨h1, h1′⟩}
                  do
                    τ̂1 ←_p (τ_p′(τ1^p))
                    if (τ̂1(h) ≤ ⟨x, y⟩[h′]) and (⟨x, y⟩[t′] ≤ τ̂1(t))
                    then if τ̂1(α) is a subshape of γ then τ_nt+←1 ← τ̂1
                  for ⟨p, p′⟩ ∈ {⟨t1, h1′⟩, ⟨h1, t1′⟩}
                  do
                    τ̂2 ←_p (τ_p′(τ2^p))
                    if (τ̂2(t) ≤ ⟨x, y⟩[h′]) and (⟨x, y⟩[t′] ≤ τ̂2(h))
                    then if τ̂2(α) is a subshape of γ then τ_nt+←1 ← τ̂2
                  γline ← next[γline]
                αline ← next[αline]
            i ← thread[N[γstack[i]]]
        L_α-node ← next[L_α-node]
      j1 ← j2
    j1 +← 1
```

Algorithm 1 (continued)

set of labelled points. The labelled shapes $\alpha - \beta$ and $\beta - \alpha$ are represented in the same manner as labelled shapes in Krishnamurti (1980). The following changes apply only to the labelled shape α.

Consider the shape $s(\alpha)$. As before, each list of colinear maximal lines is represented by a linked list which consists of, at most, two kinds of nodes: a 'header' node and 'line' nodes. The 'line' node consists of three fields: *tail*, *head*, and *next* which have the same interpretations as before. The 'header' node comes into play only if *type* = 5. That is, for a *list node* in the list representing $s(\alpha)$:

$$N[\textit{list node}] \text{ points to } \begin{cases} \text{a header node,} & \text{for } \textit{type} = 5 \\ \text{a line node,} & \text{otherwise.} \end{cases}$$

```
¶ Step 3: continued ¶
if M_α = 1
then
  ¶ All the maximal lines in α are colinear ¶
  αheader ← N[top_s[α]]
  for j ∈ ⟨1, ..., γptr⟩
  do
    γline ← top[N[γstack[j]]]
    while γline ≠ null
    do
      DP_γ[1] ← t'_1 ← tail[γline]
      DP_γ[2] ← h'_1 ← head[γline]
      αline ← top[αheader]
      while αline ≠ null
      do
        DP_α[1] ← t_1 ← tail[αline]
        DP_α[2] ← h_1 ← head[αline]
        TRANSFORMATIONS(2,2)   ¶ αline maps onto γline ¶
        αline2 ← next[αline]
        while αline2 ≠ null
        do
          t_2 ← tail[αline2]
          DP_α[2] ← h_2 ← head[αline2]
          TRANSFORMATIONS(2,2)   ¶ {tail, head} maps onto γline ¶
          γline2 ← next[γline]
          while γline2 ≠ null
          do
            t'_2 ← tail[γline2]
            DP_γ[2] ← h'_2 ← head[γline2]
            TRANSFORMATIONS(2,2)   ¶ {tail, head} maps onto {tail, head} ¶
            DP_α[1] ← h_1
            DP_γ[1] ← h'_1
            TRANSFORMATIONS(2,1)   ¶ ⟨head, head⟩ maps onto ⟨head, head⟩ ¶
            DP_α[1] ← t_2
            DP_α[2] ← t_1
            TRANSFORMATIONS(2,1)   ¶ ⟨tail, tail⟩ maps onto ⟨head, head⟩ ¶
            DP_α[2] ← h_1
            DP_γ[2] ← t'_2
            TRANSFORMATIONS(2,2)   ¶ {head, tail} maps onto {head, tail} ¶
            DP_α[1] ← h_2
            DP_γ[1] ← t'_1
            TRANSFORMATIONS(2,1)   ¶ ⟨head, head⟩ maps onto ⟨tail, tail⟩ ¶
            DP_α[1] ← t_1
            DP_α[2] ← t_2
            TRANSFORMATIONS(2,1)   ¶ ⟨tail, tail⟩ maps onto ⟨tail, tail⟩ ¶
            DP_α[2] ← h_2
            γline2 ← next[γline2]
          DP_γ[2] ← h'_1
          αline2 ← next[αline2]
        αline ← next[αline]
      γline ← next[γline]

¶ Step 4: Finishing touches ¶
if (error) or (nt = 0)
then "shape rule does not apply"
else "(τ_1, ..., τ_nt) contains all the transformations under which R applies to γ"
end SHAPE RULE APPLICATION
```

Algorithm 1 (continued)

The header node consists of three fields:

header node	H-*point*	*bot*	*top*

bot and *top* have the same interpretation as before, and H-point is a pointer to an array which stores the coordinates of points. In this case, H-*point* represents the H-intersection point of *tail*[*top*[*header node*]] and some maximal line parallel to the maximal line represented by *top*[*header node*].

The set of labelled points $P^*(\alpha)$ is represented in a similar fashion as $P^*(\gamma)$ except for the following modifications depending upon the *type* of the shape rule.

Case 1: *type* = 1 or 2. There is no change in the data structure to that described for $P^*(\gamma)$.

Case 2: *type* = 3 or 4. In this case all the labelled points of α share the same coordinates. top_P points to a node which represents either a list of ⊥-intersection points in the case that *type* = 3, or a list of end points of maximal lines of α for *type* otherwise. Since for shape rules of type 3 only one ⊥-intersection point is

Algorithm TRANSFORMATIONS(*nd*, *nm*)

¶ This procedure which is invoked by SHAPE RULE APPLICATION computes the coefficients of, and if $nm \geqslant 1$, examines for membership in $T_{R,\gamma}$, the transformation(s) defined by the mapping $DP_\alpha[j] \leftrightarrow DP_\gamma[j]$, $1 \leqslant j \leqslant nd$. When $nm = 2$, the transformations defined by the mapping $DP_\alpha[1 \dots 2] \leftrightarrow DP_\gamma[2 \dots 1]$ are also examined ¶

if $nd = 1$

then
- ¶ τ is a translation ¶
- $\tau_x \leftarrow_p \langle 0, 0, (x[DP_\gamma[1]] - x[DP_\alpha[1]])\rangle$
- $\tau_y \leftarrow_p \langle 0, 0, (y[DP_\gamma[1]] - y[DP_\alpha[1]])\rangle$
- $\tau_{nt+\leftarrow 1} \leftarrow \langle \tau_x, \tau_y\rangle$

else
- ¶ With respect to the mapping $DP_\alpha[1 \dots nd] \leftrightarrow DP_\gamma[1 \dots nd]$, construct the possible corresponding triples of points and place them in *mapping* ¶
- **case** *nd* **in**
  - (= 2) Using equations (14) through (16) compute the coordinates of the vertices of the similar triangles. Let them be stored in q_j and q'_j, $1 \leqslant j \leqslant 3$, respectively.
    - $mapping[1] \leftarrow \langle q'_2, q'_3\rangle$
    - $mapping\ [mptr \leftarrow 2] \leftarrow \langle q'_3, q'_2\rangle$
    - **if** $nm = 2$
    - **then**
      - ¶ Construct the possible triples of points corresponding to the mapping $DP_\alpha[1 \dots 2] \leftrightarrow DP_\gamma[2 \dots 1]$ ¶
      - $\langle \Delta x', \Delta y'\rangle \leftarrow_p (\langle x, y\rangle[DP_\gamma[2]] - \langle x, y\rangle[DP_\gamma[1]])$
      - $\langle x, y\rangle[q'_4] \leftarrow_p (\langle x, y\rangle[q'_2] + \langle \Delta x', \Delta y'\rangle)$
      - $\langle x, y\rangle[q'_5] \leftarrow_p (\langle x, y\rangle[q'_3] + \langle \Delta x', \Delta y'\rangle)$
      - $mapping[3] \leftarrow \langle q'_4, q'_5\rangle$
      - $mapping[mptr \leftarrow 4] \leftarrow \langle q'_5, q'_4\rangle$
  - (= 3) **for** $j \in \langle 1, 2, 3\rangle$ **do**
    - $\langle x, y\rangle[q_j] \leftarrow \langle x, y\rangle[DP_\alpha[j]]$
    - $\langle x, y\rangle[q'_j] \leftarrow \langle x, y\rangle[DP_\gamma[j]]$
  - $mapping[mptr \leftarrow 1] \leftarrow \langle q'_2, q'_3\rangle$
- ¶ Compute the coefficients of the transformation(s) and store them in $\tau^*_{1,\dots,mptr}$ ¶
- **for** $k \in \langle 1, \dots, mptr\rangle$
- **do**
  - $\langle \hat{q}'_2, \hat{q}'_3\rangle \leftarrow mapping[k]$
  - Compute the coefficients of the transformation given by the correspondence $\langle q_1, q_2, q_3\rangle \leftrightarrow \langle q'_1, \hat{q}'_2, \hat{q}'_3\rangle$ using equations (10) through (12). Let the computed transformation be $\langle \tau_x, \tau_y\rangle$
  - $\tau^*_k \leftarrow \langle \tau_x, \tau_y\rangle$
- **if** $nm > 0$
  - ¶ Examine the transformation(s) for membership in $T_{R,\gamma}$ ¶
- **then**
  - **for** $j \leftarrow 1, j$ **while** $j \leqslant mptr$
  - **do**
    - **if** $\tau^*_j(\alpha)$ is a subshape of γ
    - **then**
      - $\tau_{nt+\leftarrow 1} \leftarrow \tau^*_j$
      - **if** $type[R] = 5$
      - **then**
        - ¶ In this case $M_\alpha = 1$ (see SHAPE RULE APPLICATION) and so the mirror image of τ^*_j, τ^*_{j+1} is also in $T_{R,\gamma}$ ¶
        - $\tau_{nt+\leftarrow 1} \leftarrow \tau^*_{j+\leftarrow 1}$
    - **else if** $type[R] = 5$ **then**
      - ¶ τ^*_{j+1} is also not in $T_{R,\gamma}$ ¶
      - $j+\leftarrow 1$
    - $j+\leftarrow 1$

end TRANSFORMATIONS

Algorithm 2.

required, then only one node is pointed to by $N[top_P]$. For shape rules of type 4 only end points of maximal lines which do not share the same coordinates as the distinct labelled point need be maintained in the list pointed to by $N[top_P]$. In both cases, the nodes $next[top_P]$, $next[next[top]]$ etc represent lists of labelled points. If 'key' values of '⊥' and '•' are permitted which have the obvious interpretations, then the following ordering on the point labels is adopted: $\perp = \cdot \langle A_1 < A_2 < ... < \#$. Notice that in this case the list representing either the single ⊥-intersection point or the end points of maximal lines have no header node.

Case 3: *type* = 4 and $s(\alpha) = S_\phi$. There is no change in the data structure to that described for $P^*(\gamma)$.

Case 4: *type* = 5. In this case, top_P points to **null**.

Figure 16 presents a pictorial description of the data structures required to represent the labelled shape α.

Finally, an Algol-like translation of the shape rule application algorithm which utilizes these data structures is presented in algorithms 1 and 2.

Acknowledgements. I would like to thank George Stiny for his critical reading of an earlier version of this paper. The research reported in this paper was supported in part by a grant from the Science Research Council.

References

Krishnamurti R, 1980 "The arithmetic of shapes" *Environment and Planning B* 7 463-484

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

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

Stiny G, Mitchell W J, 1980 "The grammar of paradise: on the generation of Mughul gardens" *Environment and Planning B* 7 209-226