GEOMETRIC FRACTALS USING TRAVERSAL STRATEGIES

by

Uday G. Gujar, Virendra C. Bhavsar
Stephen Y.M. Choi and Prem K. Kalra

TR92-067 May 1992

Faculty of Computer Science
University of New Brunswick
P.O. Box 4400
Fredericton, N.B.
Canada E3B 5A3

Phone: (506) 453-4566
Fax: (506) 453-3566
Email: uday@unb.ca

TRAVERSED GEOMETRIC FRACTALS

Uday G. Gujar, Virendra C. Bhavsar,
Stephen Y. M. Choi and Prem K. Kalra

Faculty of Computer Science,
University of New Brunswick
P.O. Box 4400, Fredericton, N.B.
Canada E3B 5A3

Phone: (506) 453-4566 Fax: (506) 453-3566 Email: uday@unb.ca

ABSTRACT

In the literature it is assumed that given a line segment and a generator there is only one way to replace the line segment by the generator to produce the fractal image, i.e., a fixed traversal direction is assumed for each segment. We introduce the concept of traversal strategies wherein each of the segments of the initiator and generator can be traversed in three ways. This results in $3^M \bullet 3^Q$ *fractal images for a Q-segment initiator and an M-segment generator; these are referred to as the traversed geometric fractals. In contrast, a single fractal image is obtained without the use of traversal strategies. A software tool, Rangoli, has been developed to generate traversed geometric fractals. Several examples are included.*

Keywords: *Geometric fractals, traversal strategies, generators, initiators, fractal images.*

1. INTRODUCTION

The concept of fractals has been introduced by Mandelbrot [MAND77, MAND82]. Fractals can be classified into three groups, namely, geometric, algebraic and stochastic.

Algebraic fractals are generated from iterations on algebraic transformation functions. The fractals generated from the self-squared function $z \leftarrow z^2 + c$ have been extensively studied [MAND82, PEIT86]. Many other algebraic transformations have also been used for generating interesting fractals [PICK90, GUJA91, GUJA92a].

Stochastic fractals are fractals generated using a random process [FOUR82, VOSS85]. Coastlines and terrains are classical examples of stochastic fractals.

Fractals generated using geometric patterns are referred to as geometric fractals. These are constructed using a given generator and an initiator which are sets of arbitrary line segments. The construction process consists of replacing each segment of the initiator by the generator, reduced and displaced so as to have the end points coincide with those of the segment being replaced. Each of these generated segments is then replaced with a scaled copy of the generator. This process is repeated for the required number of iterations. In this process the given order of the end points defining the initiator and generator segments determines the segment replacement strategy and hence the final fractal image. In other words, the initiator and generator segments are traversed only in one direction defined by the given order of the end points.

In this paper, we propose the use of multiple traversal strategies for the segments of the initiator as well as those of the generator. It is shown that a large number of fractal images can be generated by changing the traversal strategies for the given initiator and/or generator. We refer to the fractals generated using the multiple traversal strategies as the *traversed geometric fractals*.

2. PRELIMINARIES

In this section we introduce notations for the initiators and generators followed by a segment identification scheme.

2.1 Initiators

An initiator is a set of line segments upon which a fractal image is built. The segments may be of equal or unequal lengths and either connected or disconnected.

An initiator is denoted by E, the i^{th} segment of an initiator is denoted by E_i and the two end points of E_i are denoted by $e_{i,1}$ and $e_{i,2}$. The number of segments in an initiator is denoted by Q. Thus, an initiator is defined as

$$E = \{E_1, E_2, \dots, E_Q\},$$

where, $E_i = (e_{i,1}, e_{i,2})$, $e_{i,j} = (x_{i,j}, y_{i,j})$ and the $x_{i,j}$ and $y_{i,j}$ are the x and y coordinates for $i \in \{1, 2, \dots, Q\}$ and $j \in \{1,2\}$.

2.2 Generators

A generator consists of a set of line segments used to generate fractal images. The line segments need not be of equal length or even connected.

A generator is denoted by G while the i^{th} segment of a generator is denoted by G_i. The two end points of G_i are denoted as $g_{i,1}$ and $g_{i,2}$. Further, the number of segments in a generator is denoted by M. Thus a generator is defined as

$$G = \{G_1, G_2, \dots, G_M\},$$

where, $G_i = (g_{i,1}, g_{i,2})$, $g_{i,j} = (x_{i,j}, y_{i,j})$ and the $x_{i,j}$ and $y_{i,j}$ are the x and y coordinates for $i \in \{1, 2, \ldots, M)$ and $j \in \{1,2\}$.

A generator has a starting point, A, and an end point, B, which are used in the generation process for a fractal image. The points A and B are denoted as Δ and $\Box$ respectively in many of the figures in this paper. Note that these may not be the same as the starting and end points of any of the generator segments. Without loss of generality the two end points of a generator are located on the X-axis at an equal distance, a, from the origin, i.e., $A = (-a, 0)$ and $B = (a, 0)$. Thus A and B give the orientation and size of a generator.

Generators can be divided into two categories, namely, symmetric and non-symmetric.

2.2.1 Symmetric generators

Symmetric generators can further be sub-divided into the following three categories: odd symmetric, even symmetric and bi-symmetric.

2.2.1.1 Odd symmetric generators

These generators exhibit symmetry through the origin. Such a symmetry is obtained by two successive reflections in mirrors placed perpendicular to the X-Y plane along the principal axes. In an odd symmetric generator, for every G_i there exists a unique G_k, for $i, k \in \{1, 2, \ldots, M\}$, such that $x_{i,1} = -x_{k,2}$, $y_{i,1} = -y_{k,2}$ and $x_{i,2} = -x_{k,1}$, $y_{i,2} = -y_{k,1}$. In other words, $g_{i,1}$ is the reflection of $g_{k,2}$ through the origin and $g_{i,2}$ is the reflection of $g_{k,1}$ through the origin.

G_i and G_k are called *odd corresponding segments*. We define a function $\Gamma_o()$ which operates on G_i to give its corresponding segment if it exists, i.e., $G_i = \Gamma_o(G_k)$ and $G_k = \Gamma_o(G_i)$. Note that when $i = k$, we have only one line segment and the mid-point of the segment lies on the origin; $\Gamma_o(G_i)$ does not exist in that case.

Some examples of odd symmetric generators are given in Fig. 1(a).

2.2.1.2 Even symmetric generators

There are two kinds of even symmetric generators, namely, even symmetric about the X-axis, and even symmetric about the Y-axis.

In an even symmetric generator about the X-axis, for every G_i there exists a unique G_k, for $i, k \in \{1, 2, \ldots, M\}$, such that $x_{i,1} = x_{k,1}$, $y_{i,1} = -y_{k,1}$ and $x_{i,2} = x_{k,2}$, $y_{i,2} = -y_{k,2}$. G_i and G_k are called *x-corresponding segments*, i.e., $G_i = \Gamma_x(G_k)$ and $G_k = \Gamma_x(G_i)$. Note that when $i = k$, we have only one line segment which lies on the X-axis and $\Gamma_x(G_i)$ does not exist in that case; i.e., for segments that lie on the X-axis, no x-corresponding segments are required. Further, a segment parallel to the Y-axis, with its mid-point on the X-axis also does not have an x-corresponding segment. Some examples of even symmetric generators about the X-axis are given in Fig. 1(b).

An even symmetric generator about the Y-axis is one where for every G_i there exists a unique G_k, for $i, k \in \{1, 2, \ldots, M\}$, such that $x_{i,1} = -x_{k,2}$, $y_{i,1} = y_{k,2}$ and $x_{i,2} = -x_{k,1}$, $y_{i,2} = y_{k,1}$. G_i and G_k are called *y-corresponding segments*, i.e., $G_i = \Gamma_y(G_k)$ and $G_k = \Gamma_y(G_i)$. Note that when $i = k$, we have only one line segment which lies on the Y-axis and in this case $\Gamma_y(G_i)$ does not exist, i.e., for segments that lie on the Y-axis no y-corresponding segments exist. Further, for segments that are parallel to the

X-axis with their mid-points lying on the Y-axis, no corresponding segments are required. Some examples of even symmetric generators about the Y-axis are given in Fig. 1(c).

2.2.1.3 Bi-symmetric generators

It is possible that a generator may be symmetric around both the X and Y axes. In that case, such a generator is odd symmetric as well as even symmetric. We refer to these generators as bi-symmetric generators. The last generator in each of the Figs. 1(a), (b) and (c) is a bi-symmetric generator.

2.2.2 Non-symmetric generators

A non-symmetric generator is a generator which is neither an odd, nor an even nor a bi-symmetric generator. Some examples of this type of generator are given in Fig. 1(d).

2.3 Segment Identification Scheme

Given a generator with M segments, there are $M!$ ways to number its segments. We introduce an identification scheme to number the segments from 1 to M. This scheme determines the starting and end points of each segment uniquely, and is crucial for traversed geometric fractals. A similar scheme is used to identify the segments of an initiator.

The algorithm for the identification scheme looks at the end-points of segments that have not yet been numbered to find the point with the smallest x coordinate. If more than one point has the same smallest x coordinate, then the point with the largest y coordinate is selected. Then the algorithm picks a second point which is connected to the first point, such that the second point has the smallest x coordinate. If more than one point has the same x coordinate, then the point with the largest y coordinate is selected. The segment with these two end-points is marked as segment 1 with the first point found being its starting point and the last point found being its end point. Similarly, other segments are located until all segments are numbered. An adjustment is needed for an even symmetric generator with x- or y-corresponding segments that are vertical in order to satisfy the definition of even symmetric generators about the X or Y axis.

Examples of how segments of generators are numbered using the identification scheme are given in Fig. 2. For the sake of clarity, some of the line segments are identified by only one of its end points. Figure 2(a) illustrates the identification of the segments and their end points for various generators. Figure 2(b) gives an even symmetric generator about the X-axis before and after the adjustment while Fig. 2(c) gives an even symmetric generator about the Y-axis before and after the adjustment. Note that no such adjustment is needed for horizontal x- or y- corresponding segments.

3. TRAVERSAL STRATEGIES

The generation process given in the literature for geometrical fractals is as follows. In the first iteration each line segment of the initiator is replaced by a copy of the generator. This copy is obtained by reducing and displacing the generator so as to have its end points coincide with those of the segment being replaced. In the second iteration this procedure of copying the generator is repeated for every line segment in the generated pattern from the first iteration. Similar process is carried out for the subsequent iterations. The termination condition is usually the preselected number of iterations. Alternatively, one can use the size of the generated segments to terminate the process. If there are M segments in the generator and Q segments in the initiator, the resulting image after N iterations consists of QM^N segments. Every level of iteration contains smaller copies of the generator exhibiting the property of exact self-similarity.

In the literature it is assumed that given a line segment, from the initiator or a generated intermediate pattern, there is only one way to replace that segment by the generator. Thus, there is a fixed *implicit* direction for each of the segments of the initiator, generator as well as the generated patterns. We refer to this as the implicit traversal direction.

We propose the notion of multiple traversal directions for each of the segments. We use vector T to encode the traversed directions of the segments of the generator. This vector T represents the traversal strategy for the generator segments.

The traversal strategy vector (T) for an M-segment generator is defined as

$$T = t_1 t_2 \ldots t_M$$

where the traversal direction of the i^{th} segment of a generator is denoted as t_i, and $t_i \in \{0, 1, 2\}$. The i^{th} segment of a generator is denoted as G_i, and the two end points of G_i are denoted as $g_{i,1}$ and $g_{i,2}$. When the traversal direction t_i of G_i equals 0, it means that G_i is being traversed from $g_{i,1}$ to $g_{i,2}$, $t_i = 1$ represents G_i being traversed from $g_{i,2}$ to $g_{i,1}$, and finally, $t_i = 2$ represents G_i being traversed from $g_{i,1}$ to $g_{i,2}$ as well as from $g_{i,2}$ to $g_{i,1}$.

An i^{th} segment, with $t_i = 2$, can be regarded as consisting of two pseudo-segments, one pseudo-segment having the traversal direction equal to 0 and another pseudo-segment having the traversal direction as 1. The total number of pseudo-segments for a generator is denoted by m. Some examples of generator traversal strategies are shown in

Fig. 3 where M, m and T for each of the generators are identified; note that the line segments are identified with only one of their end points.

Similarly, the concept of traversal strategy can be applied to the initiator. The traversal strategy of an initiator, denoted by S, defines the traversal directions of the segments in an initiator. The traversal direction of the i^{th} segment of an initiator is denoted as s_i. The traversal strategy vector (S) for a Q-segment initiator is defined as

$$S = s_1 s_2 \ldots s_Q \qquad \text{where} \qquad s_i \in \{0, 1, 2\} ,$$

The total number of pseudo-segments for an initiator is denoted by q.

For an M-segment generator, there are 3^M possible ways to traverse a generator; and for a Q-segment initiator, there are 3^Q possible ways to traverse an initiator. Thus, by using different traversal strategies for a given generator and a given initiator, one can generate $3^M \bullet 3^Q$ fractal images after N iterations without modifying the generator or the initiator. In contrast, only a single image can be obtained when the concept of traversal strategy is not used.

In the above discussion we have assumed that the traversal strategy of the generator, T, remains the same for all iterations. We refer to this as the *static traversal strategy*. It is possible to change the traversal strategy at every iteration. We denote the traversal strategy at the n^{th} iteration as T^n. This gives rise to the idea of *dynamic traversal strategy*. In this paper we explore only the effects of static traversal strategies.

4. GENERATION PROCESS

The initial fractal image, denoted by F^0, is defined to be the same as the initiator. The subsequent geometric fractal image F^1 is constructed by replacing each line segment of F^0 by a proportional copy of a generator. Each line segment of F^1 in turn can be replaced by a proportional copy of the generator to produce yet another fractal image F^2, and so on. The process of replacing each line segment of a fractal image by a proportional copy of a generator can be carried out to any desired number of iterations. A geometric fractal image at the n^{th} iteration consists of a set of line segments, and is denoted as F^n. Thus the basic principle of generating geometric fractals can be stated by the following iterative formula

$$F^n = \beta(F^{n-1}, G, T^n) ,$$

where, $\beta()$ is a function which takes a fractal image, F^{n-1}, a generator, G, and a generator traversal strategy T^n as input, and produces the fractal image corresponding to the next iteration.

An example of the traversed geometric fractal generation process is shown in Fig. 4 along with the generator and initiator used. The generator used has two connected segments of equal lengths and the initiator contains only one segment. The distance between the starting point A and the end point B of the generator is considered to be the unit length. All the segment lengths of a generator are also measured in terms of unit length; the segment lengths are denoted by $R1$ and $R2$. N represents the total number of iterations in the generation process.

Figure 5 shows the traversed fractal images produced with the nine possible traversal strategies for a two-segment generator when $N = 5$; this generator and the initiator are the same as those in Figure 4. Note that when $T = 00$ we obtain the conventional geometric fractal image reported in the literature (this image, for $N = 5$, is also given in Fig. 4).

We propose a scheme for identifying the segments of the generated fractal images based on the segment identification scheme introduced earlier. We denote each segment of the fractal image at the n^{th} iteration as ${}^{i}f^{n}_{j,k}$ where n is the iteration count, i is the pseudo-segment of E, j is the pseudo-segment of G, and k goes from 1 to m^{n-1}. Figure 6 shows an example of how segments of the generated fractal images at various iterations are numbered.

If ${}^{i}f^{n}_{j,k}$ is a direct descendant of ${}^{i}f^{n-1}_{l,p}$, then j, k, p and l are related to each other as follows:

$$p = \left\lceil \frac{k}{m} \right\rceil, \quad \text{and} \quad l = \left\lceil \frac{m(k-1)+j}{m} \right\rceil - m(p-1)\,.$$

Further, ${}^{i}f^{n-1}_{j,k}$ represents the ancestor of ${}^{i}f^{n}_{l,2(k-1)+j}$ for $1 \le l \le m$.

We denote the number of pseudo segments in the generated fractal image after n iterations as Ψ^n, and $\psi()$ as the function which returns the number of traversal directions that are equal to 2, either in S or T. Then, the number of pseudo-segments is given by

$$\begin{aligned}
\Psi^0 &= Q + \psi(S) \\
\Psi^1 &= \Psi^0 \bullet (M + \psi(T)) \\
\Psi^2 &= \Psi^1 \bullet (M + \psi(T)) = \Psi^0 \bullet (M + \psi(T))^2 ,
\end{aligned}$$

and therefore by induction

$$\Psi^n = (Q + \psi(S)) \bullet (M + \psi(T))^n.$$

5. EXAMPLES

We have developed a software package, name Rangoli, to generate traversed geometric fractal images [GUJA90]. Rangoli, a word in Marathi (a language from the Maharashtra state of India), refers to the beautiful geometric patterns drawn in the frontyard of a house. Rangoli is written in VS FORTRAN and runs on an IBM 3090-

180VF. The system runs under MVS/XA TSO using the IBM 5080 graphics workstation and uses a subset of the IBM supplied graPHIGS software package [IBM86] as the host graphics environment system. The figures given in this section are generated using Rangoli. In each of these figures, the distance between the starting point A and the end point B of the generator is considered to be of unit length. All the segment lengths of the generator are measured in terms of this unit length are included as Ri, for $i = 1, 2, \dots, M$.

Fractal images from odd symmetric generators with single and multiple segment initiators are given in Fig. 7. The generator in Fig. 7(a) has three connected segments of equal lengths and these segments are connected to the starting points A and the end point B. Four disconnected segments with unequal lengths make up the generator in Fig. 7(b); the segment G_1 ends at A while the G_4 starts B. Figure 7(c) shows examples of fractal images generated with a connected multi-segment initiator and the same generator as that given in Fig. 7(a). The multi-segment initiator in Fig. 7(d) has three disconnected segments of unequal lengths, while the generator has three connected segments which are not connected to A or B. From these figures it is apparent that different traversal strategies of generators and initiators produce the same fractal image for an odd symmetric generator.

The even symmetric generator used in Fig. 8 has symmetry around the Y-axis and consists of three connected segments of equal lengths. It is possible to generate ($3^3 \bullet 3^1$=) 81 images by selecting all the possible traversal strategies for the initiator and the generator. Figure 8 shows (3^3=) 27 images obtained by choosing $s = 0$ and selecting all the possible values for T. It is seen that a different image is obtained by a change in T.

An even symmetric generator about the X-axis with six connected segments of unequal lengths, connected to A and B, is used in Fig. 9. Note that only four out of possible (3^6=) 729 combinations for S = 021 are shown. We see that interesting images can be generated with just two iterations when the generator has many segments. The number of pseudo-segments for T = 000000 is given as

$$\begin{aligned} \Psi^2 &= (Q + \psi(S)) \bullet (M + \psi(T))^2 \\ &= (3 + 1) \bullet (6 + 0)^2 \\ &= 144 \end{aligned}$$

This can be easily verified by inspecting the figure; note that some of the pseudo-segments overlap. For example, in the image generated from the initiator segment 1, we have 36 pseudo-segments due to six pentagons (each composed of six segments since M = 6) with some overlapping segments. Similarly, for T = 111111 and T = 001100, each of the images consists of 144 pseudo-segments. For T = 222222 the number of pseudo-segments in the image is given by

$$\Psi^2 = (3 + 1) \bullet (6 + 6)^2 = 576.$$

Since many segments are overlapping it is difficult to count them.

Interesting images generated with a bi-symmetric generator and a multi-segment initiator are shown in Fig. 10. The symmetry in these images resembles patterns that one sees through a kaleidoscope. The number of pseudo-segments for N = 1 is equal to 40 while for N = 2 it is 200.

The generator shown in Fig. 11 consists of four connected segments. Although the generator appears to be symmetric, it is actually non-symmetric due to the positions of points A and B. The number of pseudo-segments for T = 0000, 0010 and 1010 is 128 while that for T = 2222 is 512.

We have carried out extensive experimentation using Rangoli as a tool for investigation and the results from these experiments are stated as properties [CHOI89, GUJA92b].

6. CONCLUSION

In this paper we have formulated notations for geometric fractals. The generators are classified into four categories namely odd symmetric, even symmetric, bi-symmetric and non-symmetric. Even symmetric generators are further subdivided into even symmetric about the X or Y axis. Segment indentification schemes for generators, initiators and generated fractal images are given. The concept of traversal strategies for the initiator and generator has been introduced. It allows the generation of $3^M \bullet 3^Q$ fractal images from a given M segment generator and Q segment initiator for a fixed number of iterations, in contrast to a single image that can be obtained otherwise. A software tool, named Rangoli, has been designed and developed to generate traversed geometric fractals. Several examples of traversed fractals generated from various classes of generators and single and multi segment initiators have been given. It is found that the traversed fractal images from odd symmetric generators with different traversal strategies are identical (although the number of pseudo-segments in each image may be different).

REFERENCES

[CHOI89] Choi, S.Y.M., 'Geometric Fractals', M.Sc.(C.S.) Thesis, School of Computer Science, University of New Brunswick, Fredericton, N.B., Canada, 1989.

[FOUR82] Fournier, A., Fussel, D. and Carpenter, L.C., 'Computer rendering of stochastic modes', Comm. of ACM, Vol. 25, No. 6, pp. 371-384, 1982.

[GUJA90] Gujar, U.G., Bhavsar, V.C., and Choi, Y.M.C., 'Rangoli', Proc.. APICS Annual Computer Science Conference, Universitè de Moncton, Moncton, N.B., Canada, pp. 1-15, Nov. 1990.

[GUJA91] Gujar, U.G. and Bhavsar, V.C. 'Fractals from $z \leftarrow z^\alpha + c$ in the complex c-plane' Computers and Graphics, Vol. 15, No. 3, pp. 441-449, 1991.

[GUJA92a] Gujar, U.G., Bhavsar, V.C., and Vangala, N., 'Fractal images from $z \leftarrow z^{\alpha} + c$ in the complex z-plane', Computers and Graphics, Vol. 16, No. 1, pp. 45-49, 1992.

[GUJA92b] Gujar, U.G., Bhavsar, V.C., and Choi, S.Y.M., 'Properties of traversed geometric fractals', in preparation.

[IBM86] IBM, 'Programer's Reference graPHIGS', SC33-8104-02, Sept. 1986.

[MAND77] Mandelbrot, B.B., 'Fractals: Form, Chance, and Dimension', W.H. Freeman Company, San Francisco, 1977.

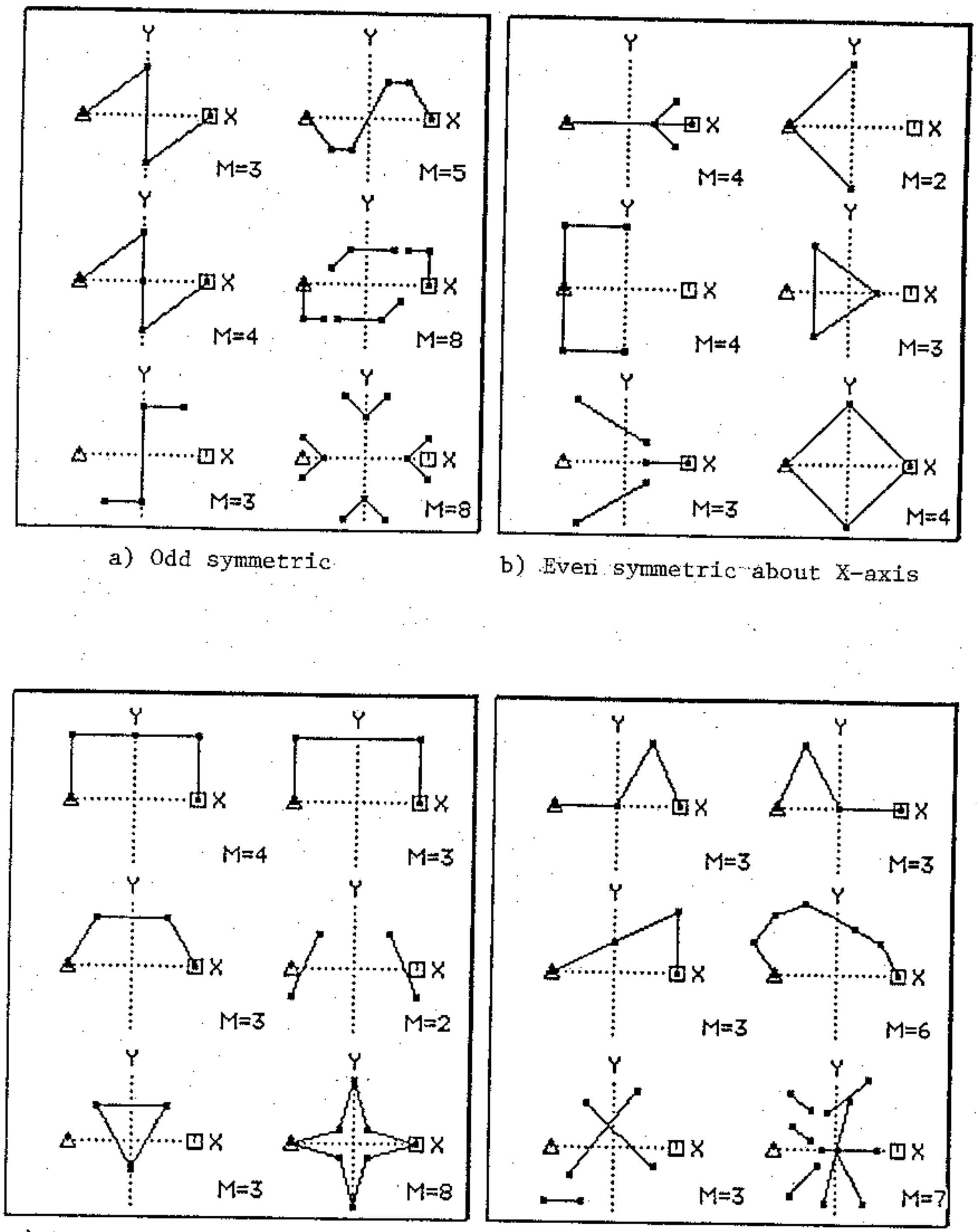
[MAND82] Mandelbrot, B.B., 'The Fractal Geometry of Nature', W.H. Freeman Company, San Francisco, 1982.

[PEIT86] Peitgen, H.O., and Richter, P.H., 'The Beauty of Fractals: Images of Complex Dynamical Systems', Springer-Verlag, Berlin, 1986.

[PICK90] Pickover, C.A., 'Computers, Pattern, Chaos and Beauty', St. Martin's Press, New York, 1990.

[VOSS85] Voss, R.F., 'Random fractal forgeries', Course notes on 'Fractals', SIGGRAPH'85, San Francisco, CA, 1985.

UGG-56



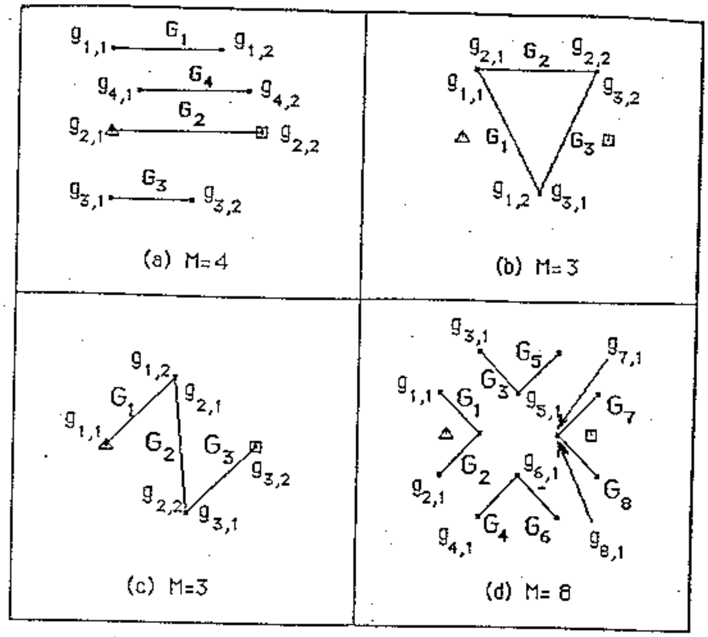
a) Odd symmetric

b) Even symmetric about X-axis

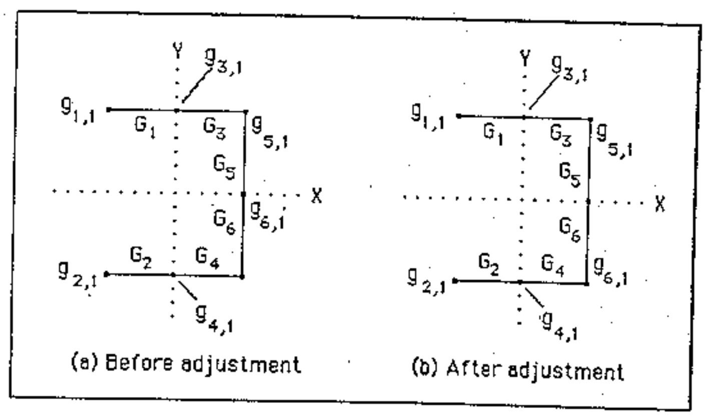
c) Even symmetric about Y-axis

d) Non-symmetric

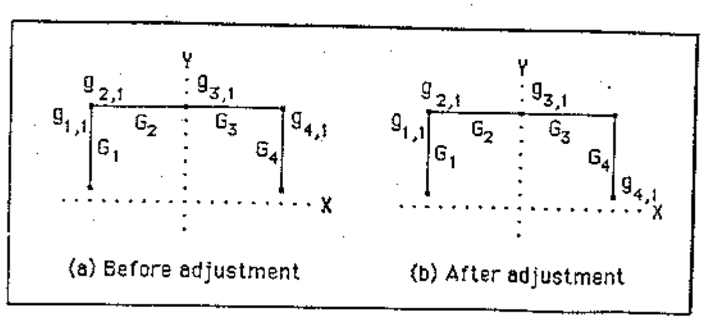
Fig. 1. Examples of generators.



a) Generators



b) Adjustment for an even symmetric generator about X-axis



c) Adjustment for an even symmetric generator about Y-axis

Fig. 2. Segment identification.

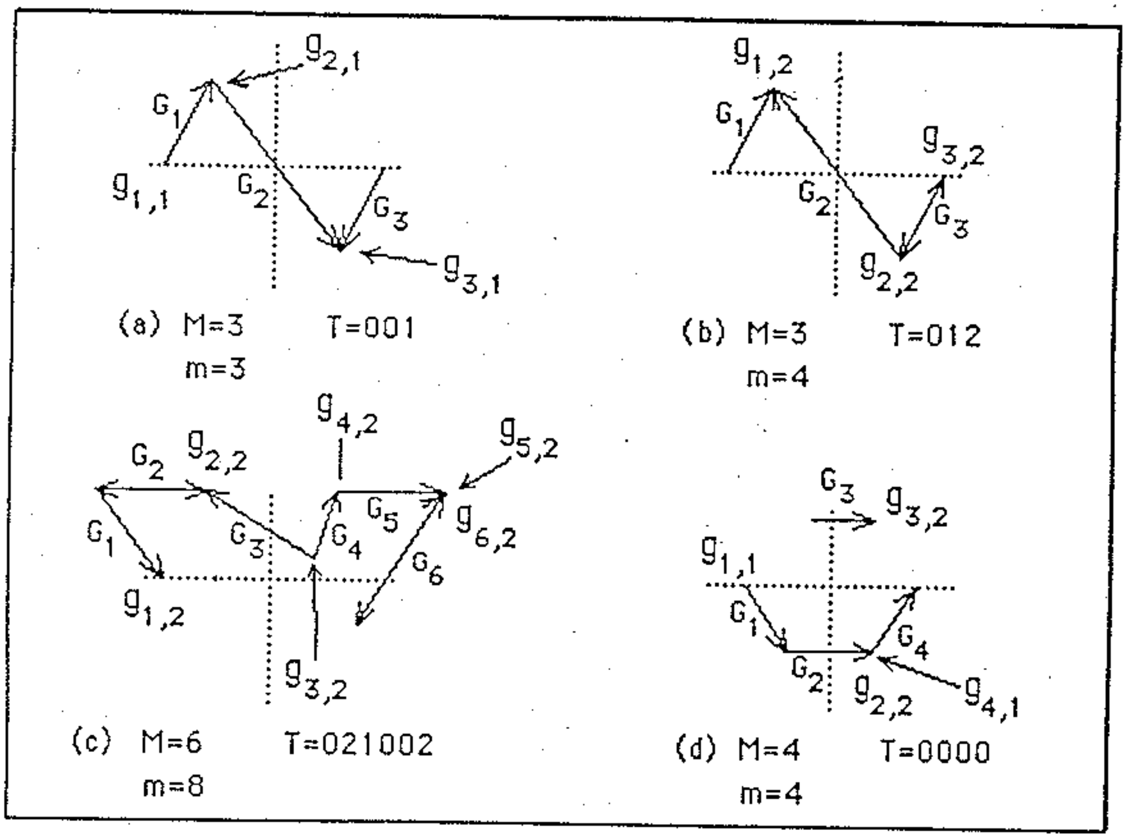


Fig. 3. Traversal strategies.

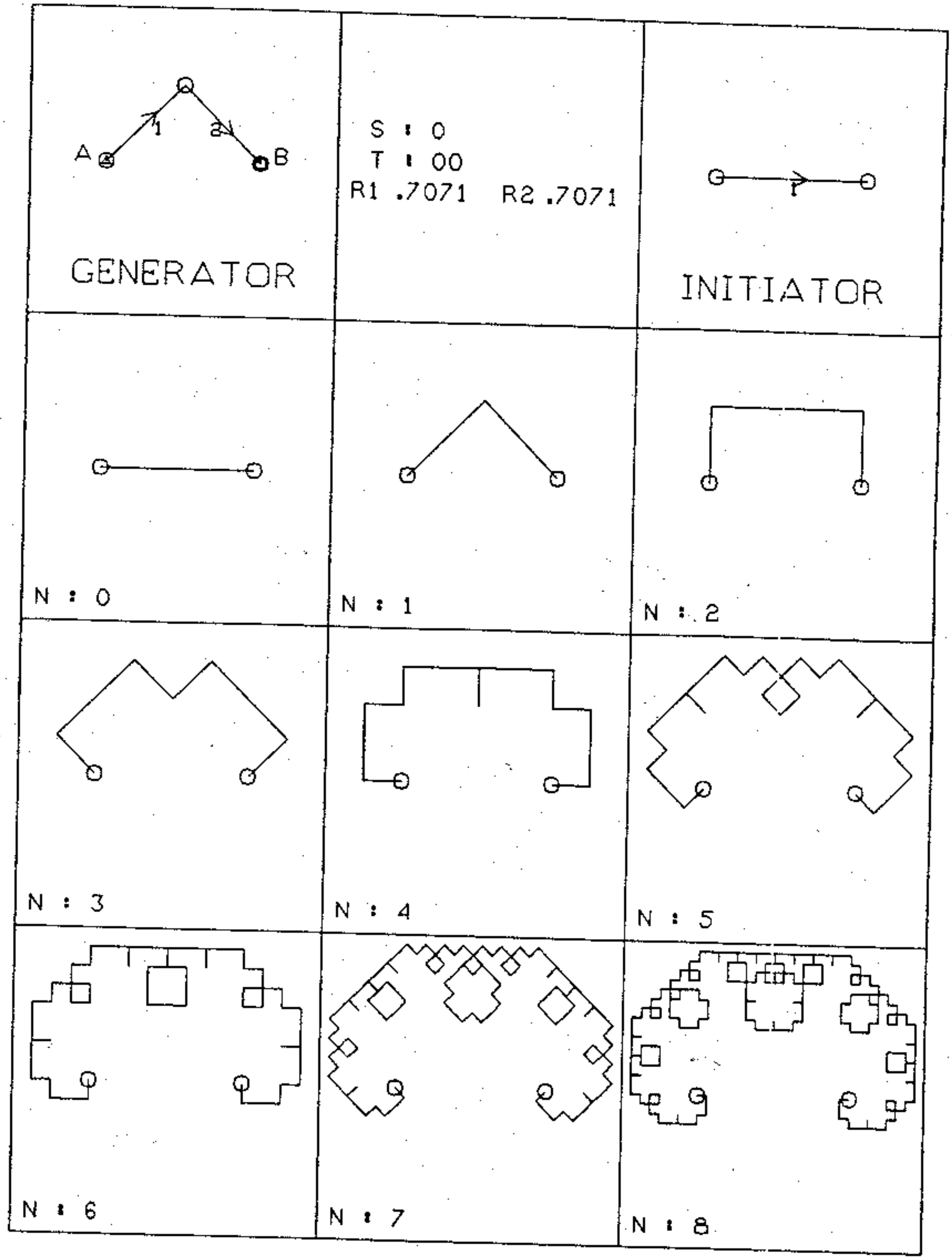


Fig. 4. Traversed geometric fractal generation process.

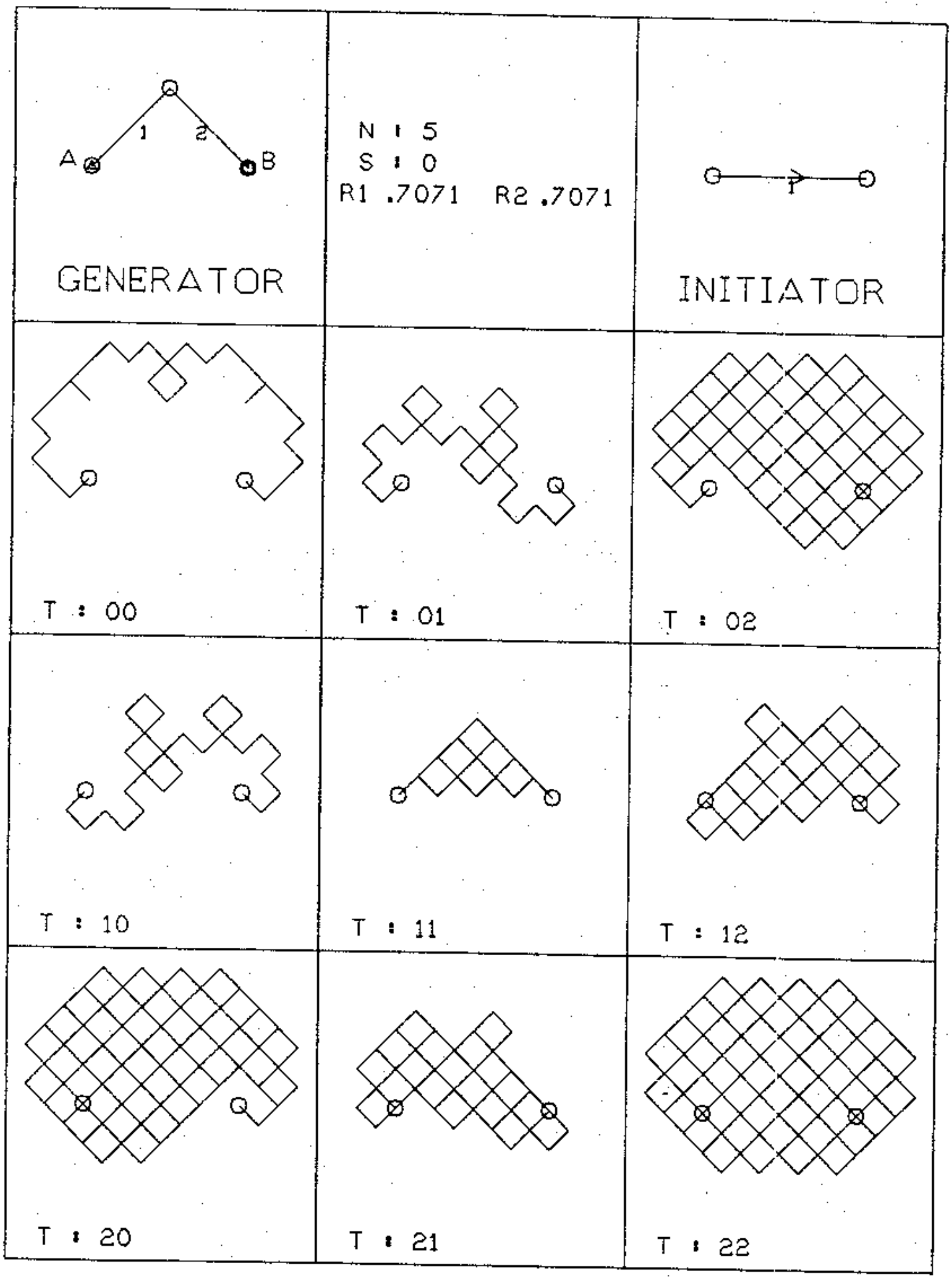


Fig. 5. Traversed geometric fractal images.

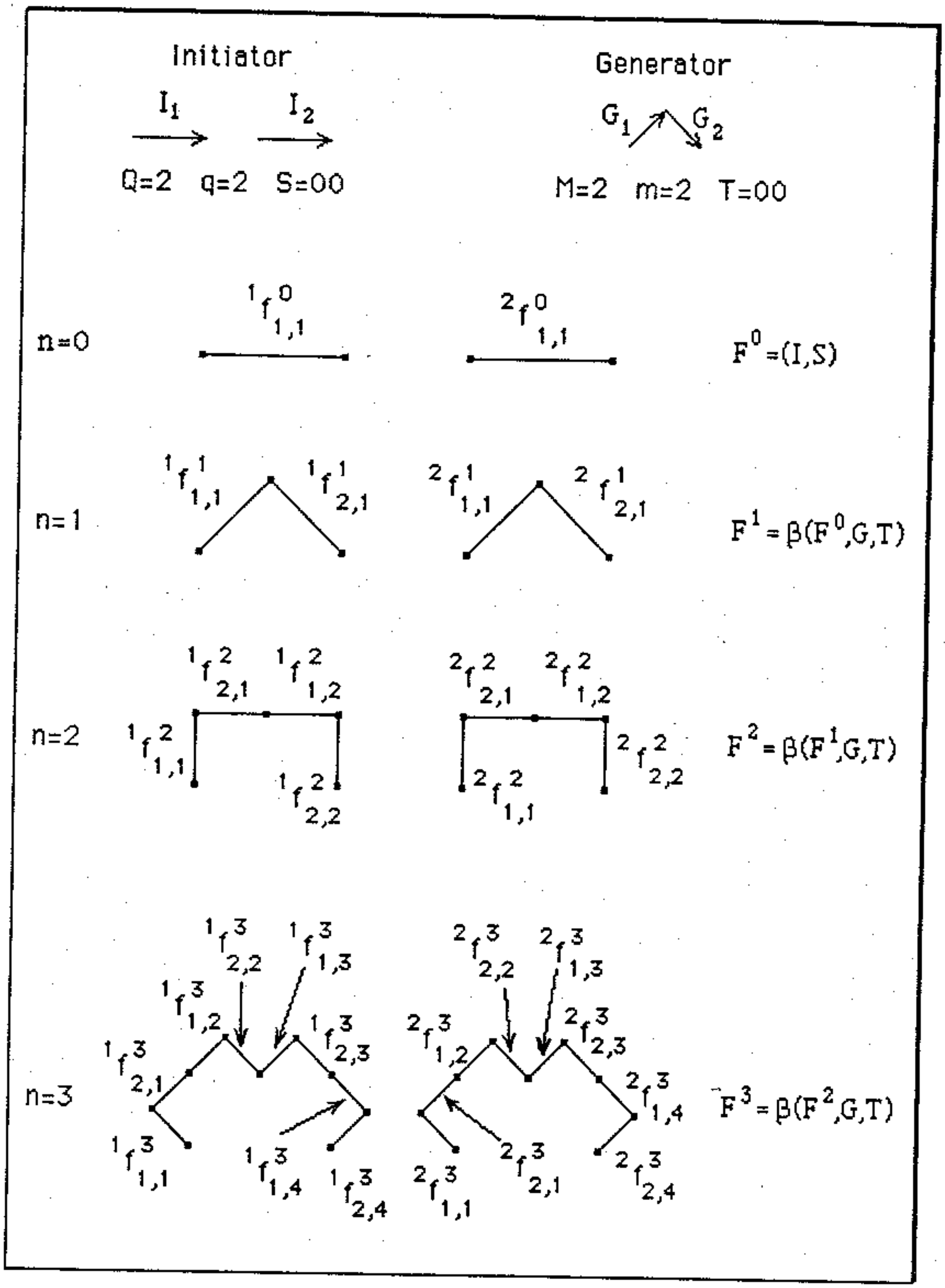
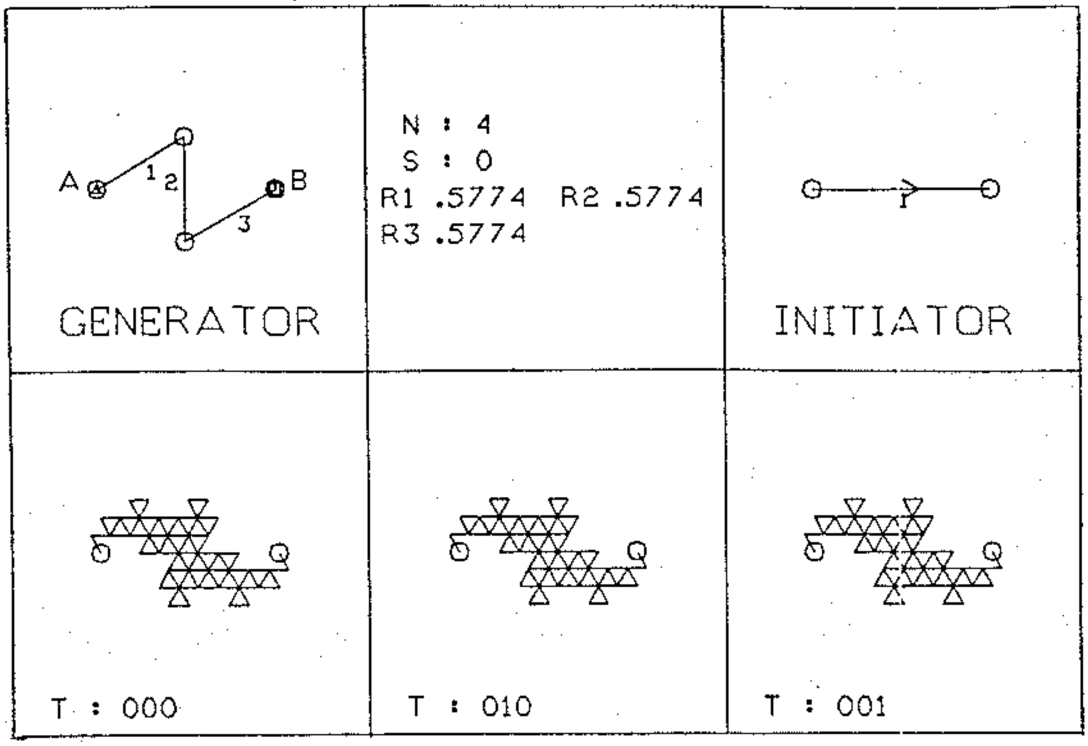
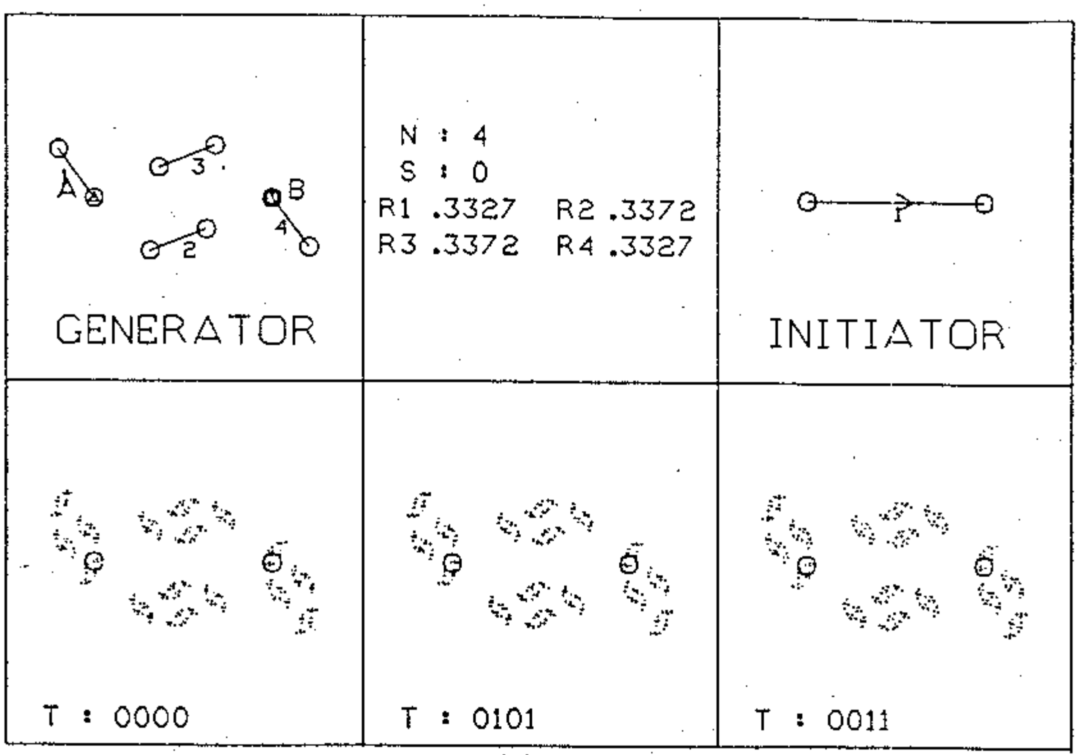


Fig. 6. An example of the number of fractal segments.

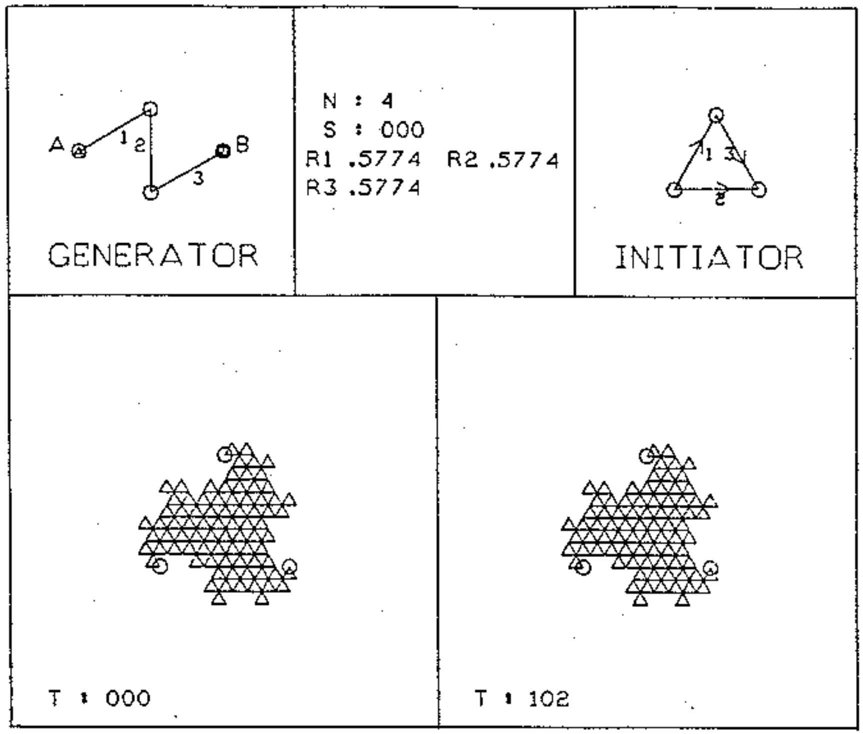


(a)

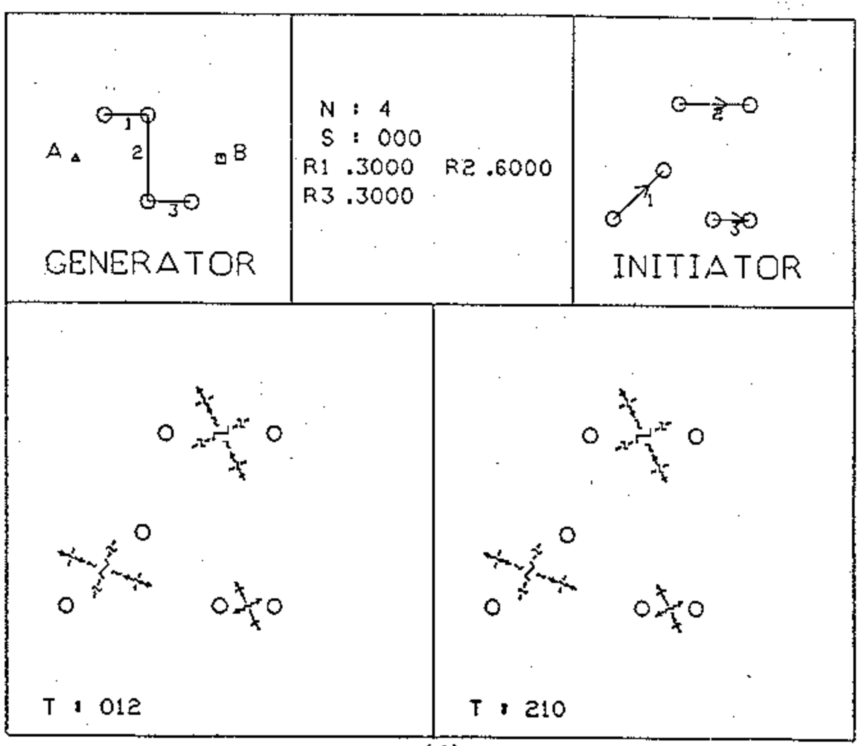


(b)

Fig. 7. Fractal images from odd symmetric generators.

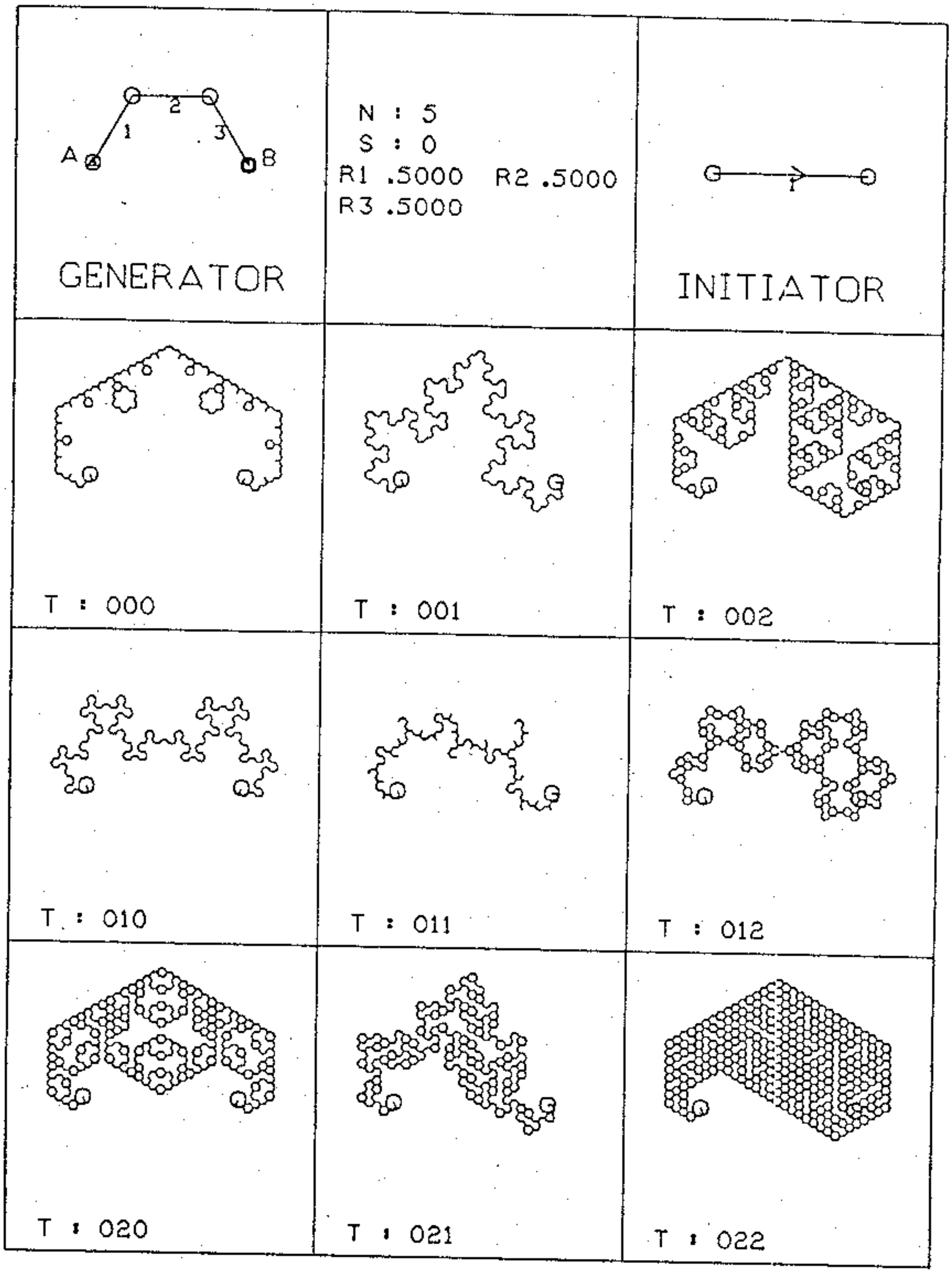


(c)



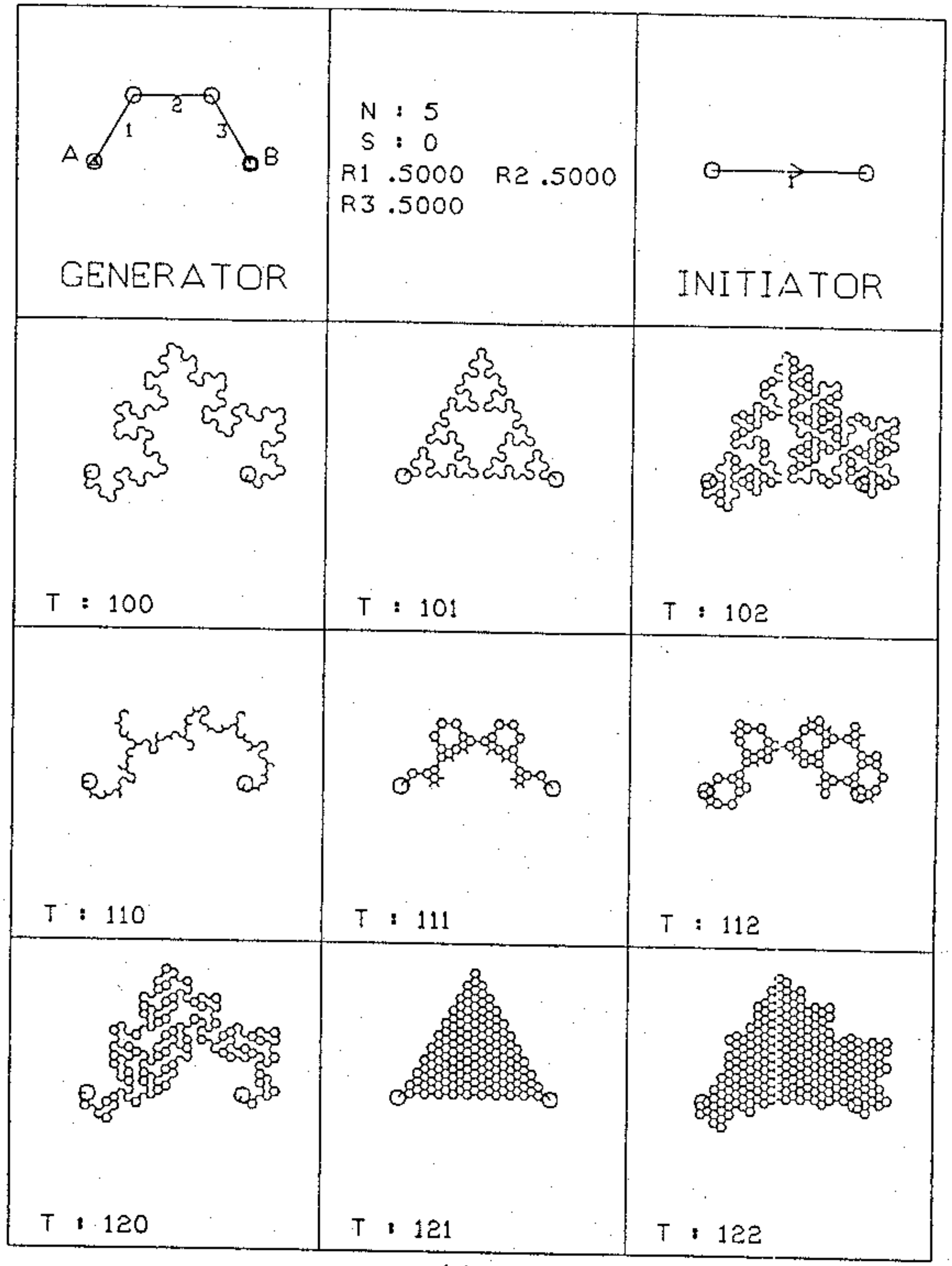
(d)

Fig. 7. Fractal images from odd symmetric generators (contd.).



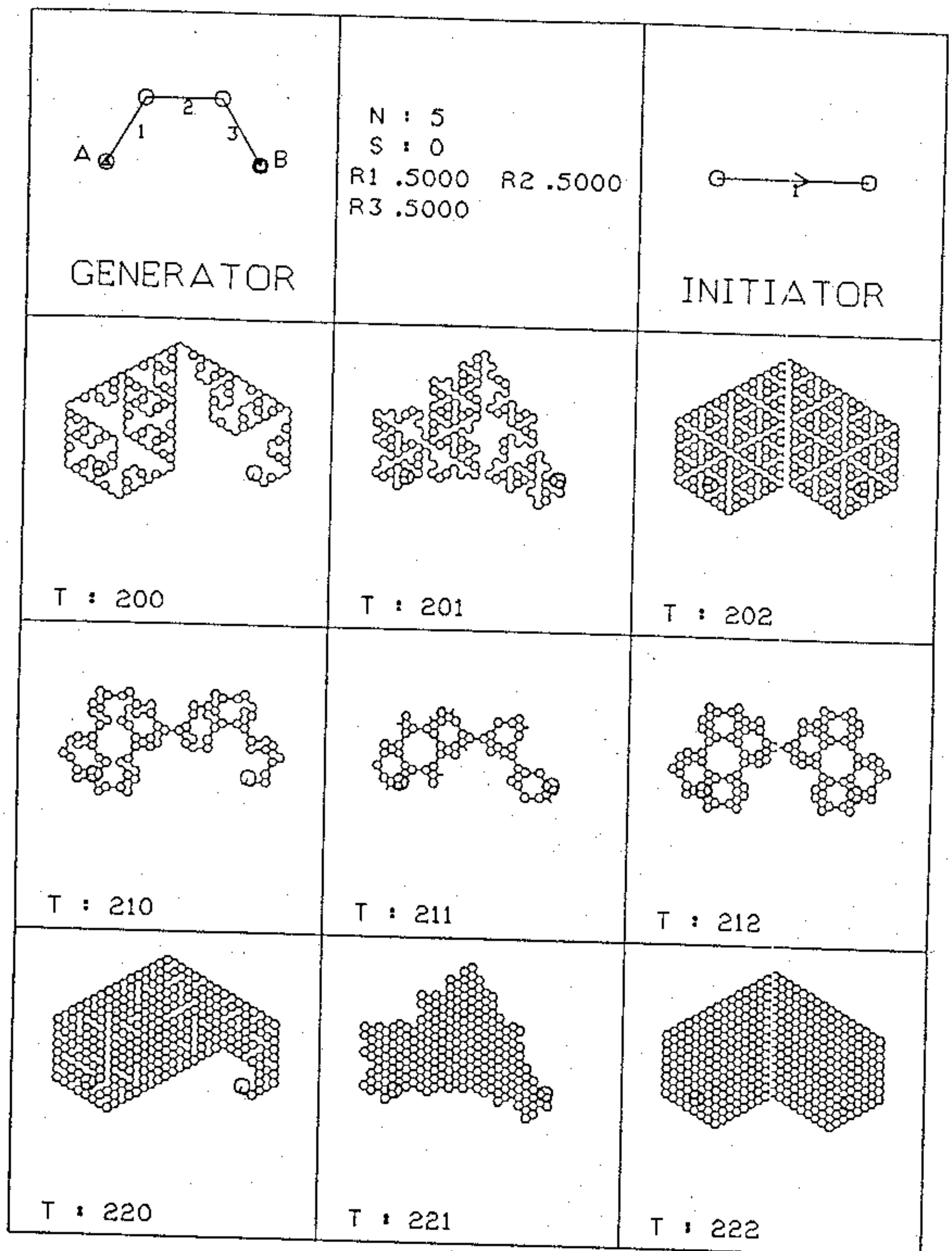
(a)

Fig. 8. Fractal images from an even symmetric generator about Y-axis.



(b)

Fig. 8. Fractal images from an even symmetric generator about Y-axis (continued).



(c)

Fig. 8. Fractal images from an even symmetric generator about Y-axis (continued).

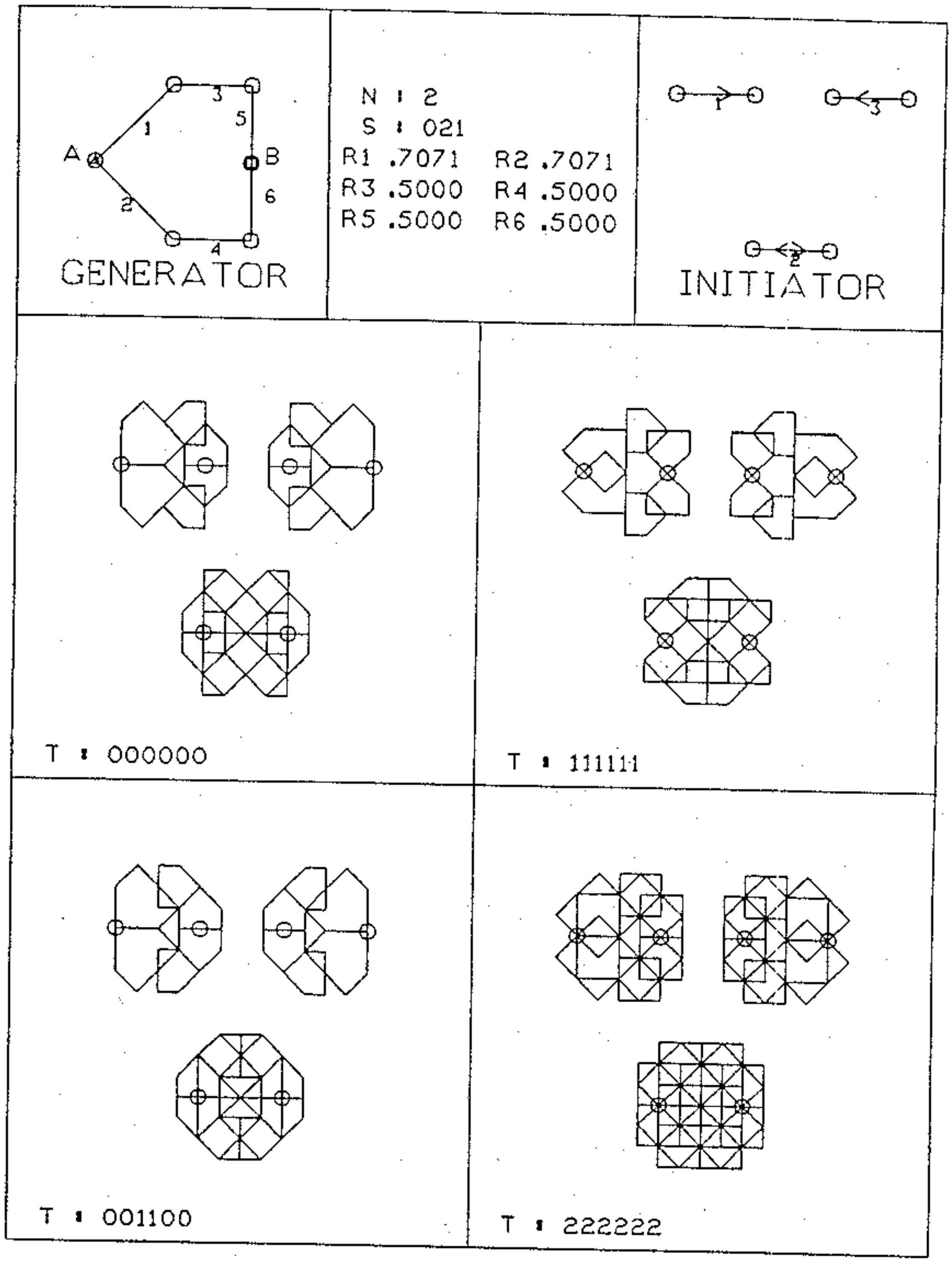


Fig. 9. Fractals from an even symmetric generator about X-axis.

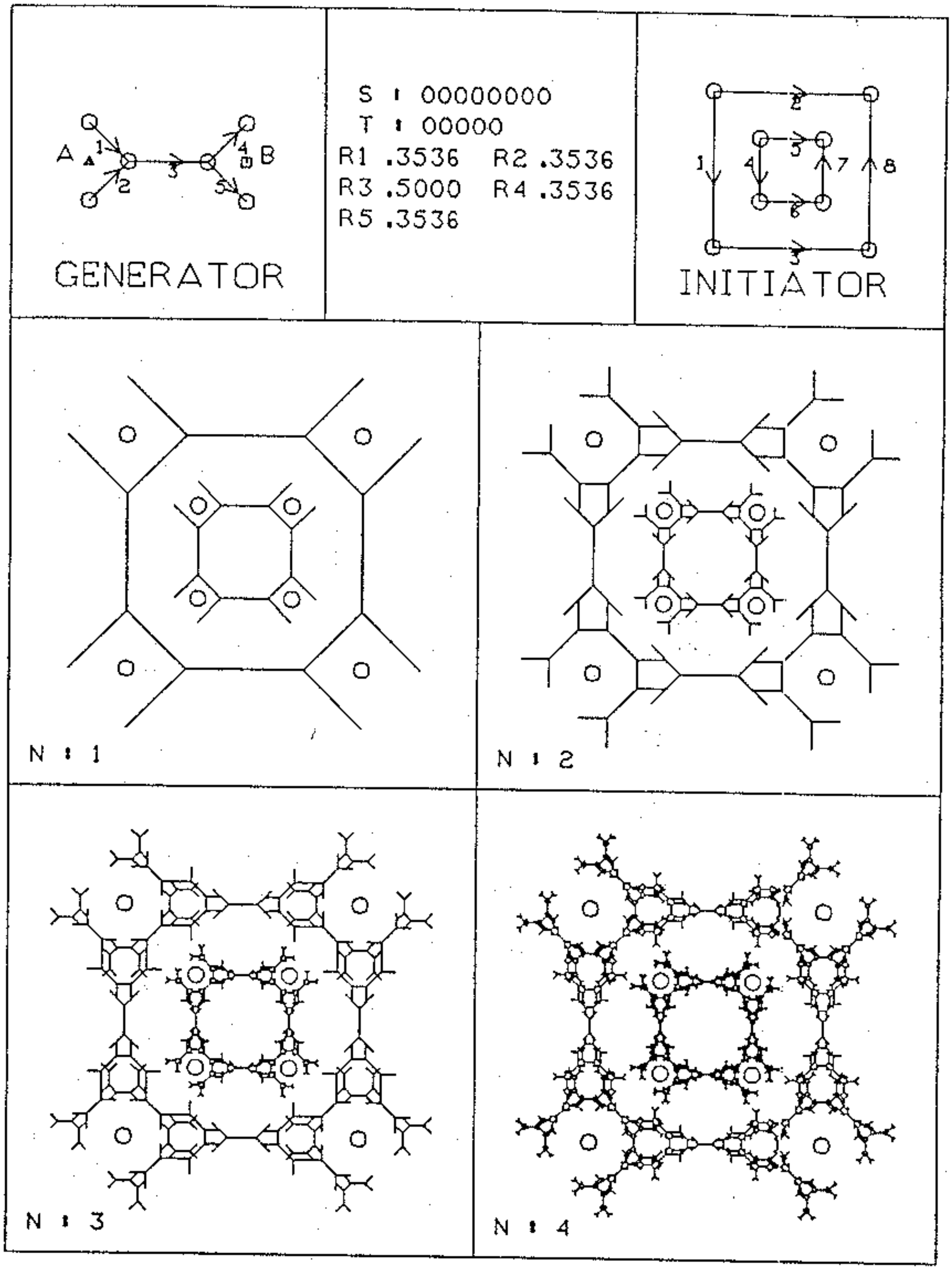


Fig. 10. Fractals from a bi-symmetric generator.

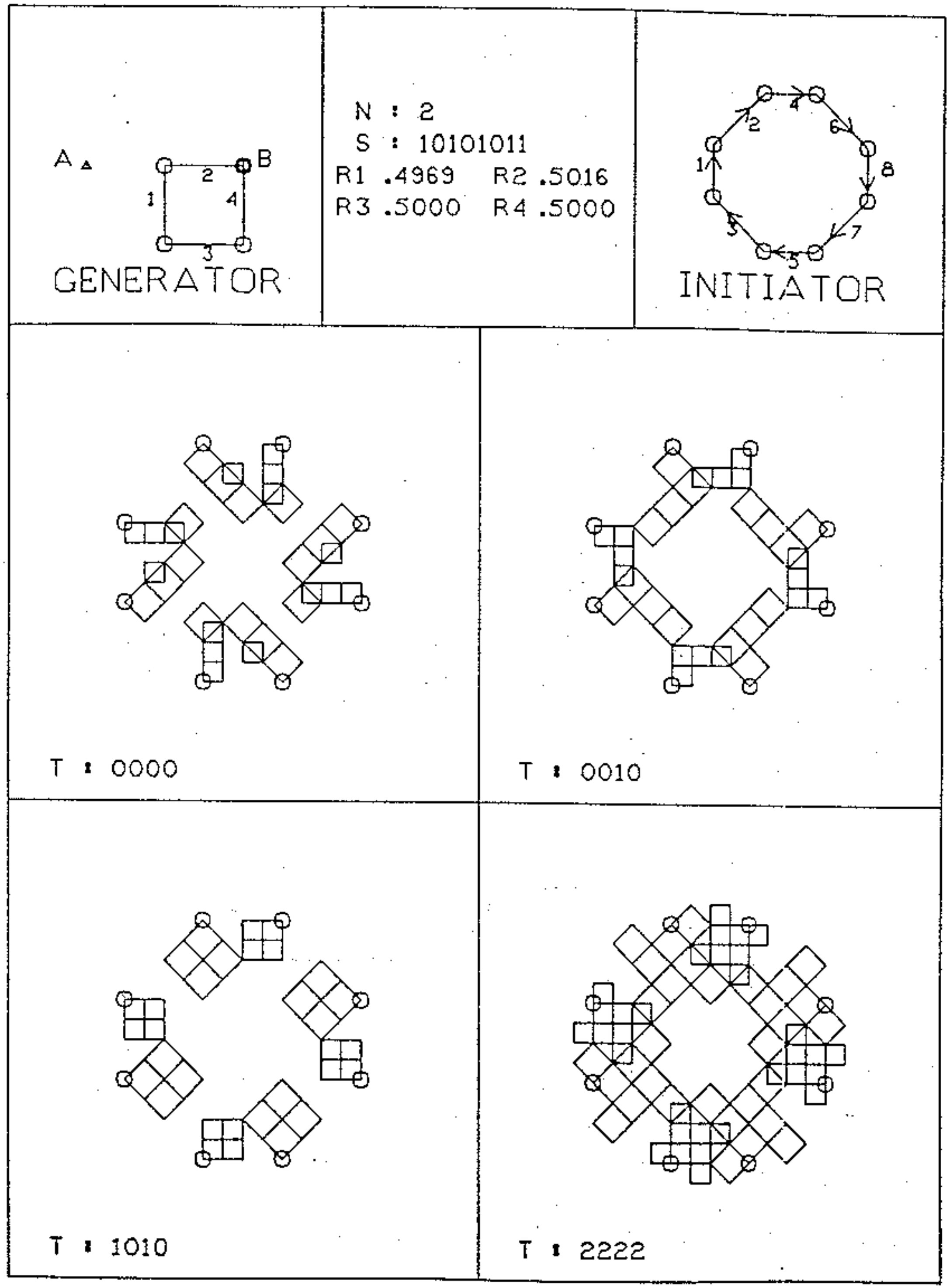


Fig. 11. Fractals from a non-symmetric generator.