Science on Form: Proceedings of the First International Symposium for Science on Form,
General Editor: S. Ishizaka, Editors: Y. Kato, R. Takaki, and J. Toriwaki, 471–477.

Nonperiodic Tesselation with Eight-fold Rotational Symmetry

Y. Watanabe, M. Ito, T. Soma and T. Betsumiya*

The Institute of Physical and Chemical Research, Wako-shi, Saitama 351-01, Japan
**Unique Design Studio, 1-9-6 Yoyogi, Shibuya-ku, Tokyo 151, Japan*

Keywords: nonperiodic tesselation, 8-fold symmetry, self similar, square, rhombus, quasi-lattice

Nonperiodic tesselation with 8-fold rotational symmetry is obtained by a self-similar subdividing operation of two kinds of cells a rhombus and a square. Each unit cell is derived from dividing the regular octagon which is uniquely divided into 16 rhombi and 8 squares with 8-fold symmetry at the center. The original unit of the tesselation is divided into 14 rhombi and 10 squares in the first subdividing operation which is called the basic unit operation. The tesselation after n times self-similar subdividing operations of the basic unit is called n-th generation pattern. The ratio of the number of squares to that of rhombi in each generation varies with n converging to $\sqrt{2}/2$. This irrational value proves the nonperiodicity of the pattern. The diffraction pattern of the tesselation is computed and shown to have sharp Bragg-like peaks with 8-fold symmetry, which might prove that the tesselation is 2-D quasi-lattice.

INTRODUCTION

Nonperiodic tesselation with 5-fold rotational symmetry is well known as the Penrose pattern(Penrose(1974)). de Bruijn(1981) derived the two kinds of rhombic cells from algebraic theory and gave the concept of quasi-lattice. Mackay(1982) and Ogawa(1985) expanded the 2-D Penrose tile to a 3-D model which consists of two kinds of rhombohedra, and elucidated the constitutions of 2-D/3-D nonperiodic structure by the recursion rule for subdividing the two kinds of cells(rhombi(2-D) and rhombohedra(3-D)). The diffraction pattern is obtained using optical transform by Mackay(1982) and computed based on mathematical treatment by Levine & Steinhardt(1984). Since Shechtman et al(1984) discovered the electron diffraction spots with 10-fold symmetry for rapidly cooled Al_6Mn alloy, many scientists (Bancel et al(1985), Kalugin et al(1985), Kimura et al(1985), Hiraga et al(1985)) carried out the diffraction experiments of various composition of Al-Mn alloy and observed a remarkably similar diffraction pattern to the calculated one by Levine & Steinhardt. The discovery of nonperiodic structure in the materials such as Al-Mn alloy made an impact on crystallographers and proved the new concept of quasi-crystal proposed previously by Mackay(1981) and de Bruijn(1981). It is an open question whether there exists the quasi-lattice which shows diffraction pattern with higher order of symmetry.

We tried to generate nonperiodic tesselation with 8-fold symmetry. In this paper we show that the tesselation can be generated by a self-similar subdividing operation of two kinds of cells which are derived from dividing the regular octagon into 16 rhombi and 8 squares. Also we found a self-similar subdividing operation for the original unit composed of the two kinds of cells as in the Mackay's rule. In as far as we know the 8-fold symmetry diffraction pattern of the nonperiodic structure has not yet been found, so we try to compute the diffraction pattern of this tesselation with 8-fold symmetry.

PATTERN GENERATION

1. Generation of nonperiodic tesselation with 8-fold symmetry

Kramer & Neri(1984) and Duneau & Katz(1985) formulated the Penrose pattern as a projection of the cubic n-grid of a higher dimensional space to a lower one. The self-similar subdividing operation of rhombic/square cells which constitute the regular octagon is another method of generating the nonperiodic tesselations as is shown in this study.

A regular octagon with 8-fold symmetry can be uniquely subdivided into 16 rhombi and 8 squares(Fig. 1 a). An assembly of one rhombus and one square in which one of the edges of these is shared is chosen as an original unit of the tesselation as shown by bold lines (Fig. 1 a). The original unit is subdivided into 14 rhombi and 10 squares (Fig. 1 b). The way of subdivision is called the basic unit of the self-similar operation. Therefore the original unit (Fig. 1 a) is the 0-th generation pattern, and the subdivided one (basic unit)(Fig. 1 b) is the 1st generation pattern. We can obtain 2nd, 3rd,... generation patterns by the succeeding operation. This self-dividing rule is derived from the regular octagon (Fig. 2 a and Fig. 2 b). The 1st generation pattern of the square has rhombic assembly with 8-fold symmetry in its center and that of the acute rhombus includes a regular sub-octagon which consists of 2 squares and 4 rhombi as shown by the shaded area (Fig. 1 b), with mirror symmetry. Therefore eight different tesselations could be derived using the basic pattern obtained by rotating the sub-octagon about its center. The ratio of similarity from n-th to (n+1)-th generation for this tesselation is $1/(2+\sqrt{2})$. All the coordinates of lattice points in the n-th generation pattern can be computed from those of the 0-th generation pattern by applying the self-similar subdividing operation recursively.

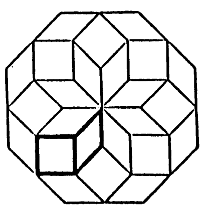

Fig. 1 a Divided regular octagon

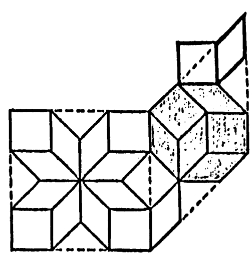

Fig. 1 b Basic unit pattern

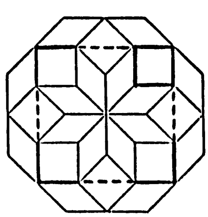

Fig. 2 a Selfdividing rule in square cell

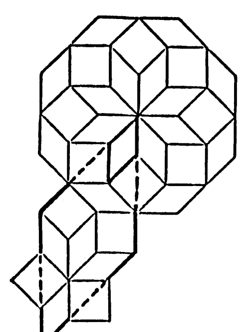

Fig. 2 b Selfdividing rule in rhombic cell

2. Proof of nonperiodicity

Mackay(1981) showed that in the pentagonal pattern the ratio of the numbers of the two kinds of cells (acute and obtuse rhombus) converges to τ the golden number. The convergency of this ratio is investigated for octagonal tesselation. The ratio of the number of squares to rhombi for the n-th generation with n tending to infinity is obtained by solving the recurrence formula for the number of squares and rhombi. Let S_n be the number of squares and R_n be the number of rhombi in the n-th generation. Then the recurrence formula written in matrix form is

$$\mathbf{X}_n = A \cdot \mathbf{X}_{n-1} = A^n \cdot \mathbf{X}_0 . \tag{1}$$

Here $\mathbf{X}_n$ is a column vector given by

$$\mathbf{X}_n = \begin{bmatrix} S_n \\ R_n \end{bmatrix} \tag{2}$$

with S_0=1 and R_0=1 , and A is a matrix

$$A = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \tag{3}$$

with a_{11} = 6, a_{12} = 8, a_{21} = 4, a_{22} = 6, After simple calculation we obtain A^n

$$A^n = \begin{bmatrix} \frac{1}{2}\{(6+4\sqrt{2})^n + (6-4\sqrt{2})^n\} & \frac{\sqrt{2}}{2}\{(6+4\sqrt{2})^n - (6-4\sqrt{2})^n\} \\ \frac{\sqrt{2}}{4}\{(6+4\sqrt{2})^n - (6-4\sqrt{2})^n\} & \frac{1}{2}\{(6+4\sqrt{2})^n + (6-4\sqrt{2})^n\} \end{bmatrix} . \tag{4}$$

The ratios of the number of squares to that of rhombi in the n-th generation in a square s_n, that in a rhombus r_n, and that in a pattern of both added t_n, respectively are expressed as

$$s_n = \frac{a_{11}^{(n)}}{a_{12}^{(n)}} \tag{5}$$

$$r_n = \frac{a_{21}^{(n)}}{a_{22}^{(n)}} \tag{6}$$

$$t_n = \frac{a_{11}^{(n)} + a_{21}^{(n)}}{a_{12}^{(n)} + a_{22}^{(n)}} . \tag{7}$$

Substituting the matrix element of eq.(4) into (5), (6) and (7), and letting n approach to infinity then the ratios s_n, r_n and t_n converge to $\sqrt{2}/2$. This proves that the octagonal tesselation pattern is nonperiodic as in the case of pentagonal tesselation (Mackay(1981)).

3. Diffraction pattern of quasi-lattice

The diffraction pattern of the nonperiodic tesselation was calculated assuming that point-like atoms are located at lattice points of the tesselation. In this case the diffraction intensity of the atoms is given by

$$I = \sum_{n,m} \exp\{2\pi i(\mathbf{s}-\mathbf{s}_0)(\mathbf{r}_n-\mathbf{r}_m)/\lambda\} \tag{8}$$

or

$$I(\mathbf{Q}) = \int g(\mathbf{r})\exp(2\pi i\mathbf{Q}\mathbf{r})dxdy, \tag{9}$$

where $g(\mathbf{r})$ is the 2-D number density function of the atom pairs, and $\mathbf{Q}=(\mathbf{s}-\mathbf{s}_0)/\lambda$. There have been proposed an elegant method of the Fourier transformation using projection technique (Duneau et al(1985), Elser(to be published), Kalugin et al(1985)). However, we adopted the direct Fourier transform of the finite number of atoms using FFT (Fast Fourier Transform) algorithm, and this method could be applied to any nonperiodic tesselation.

$I(\mathbf{Q})$ of the present octagonal pattern(Fig. 3 a) containing 1661 atoms was calculated, and the result is shown in Fig. 4 b. In Fig. 3 b we can see sharp Bragg-like peaks with 8-fold symmetry.

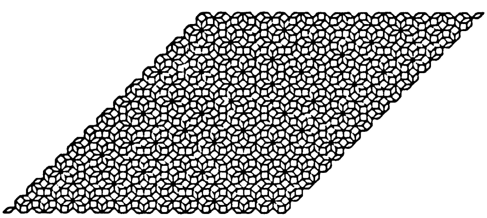

Fig. 3 a Octagonal lattice in rhombus cell

Fig. 3 b Diffraction pattern of octagonal lattice

4. Pattern generation program

A program was prepared for drawing the tesselation with 8-fold rotational symmetry. The algorithm is best described using recursive function definition because of the recursive nature of the self-similar subdividing operation. A FORTRAN program is shown in APPENDIX-2 in which the recursion is simulated by providing the stack for return address and local variables as arrays. The function EIGHT() (actually a piece of code in the main program) has five parameters: ALEN to specify the diagonal of the square or the rhombus to draw, ANG to specify the angle the diagonal makes to horizontal axis, ITY to select the rhombus or the square to draw, ISN to specify upper or lower half drawing in the case of the square (for the rhombus a complete pattern is drawn), and IPA to specify pattern filling or boundary drawing.

The function is defined recursively and stops if the diagonal becomes less than PMINL in the case of the rhombus and PMINL*RS in the case of the square, RS being the ratio of the diagonal of the square to that of the rhombus of equal side length. PMINL is calculated from the generation number IGEN and the ratio of similarity R1. IGEN together with PLENG and PANGLE is defined as input parameter. PLENG and PANGLE are passed to the function EIGHT() as ALEN and ANGLE. Since this program traverses the same lattice points several times care must be exercised to get rid of the cases in which the same points have different integer coordinates due to round-off error (add 0.51 for rounding and use double precision real variables for x and y coordinates). The coordinates of lattice points (which are necessary for calculating diffraction pattern) can be collected by providing a subroutine like SETARY() which sets the current xy-coordinate of the drawing pen with respect to the absolute origin but means are needed to reject duplications.

CONCLUSION

The division of a regular octagon into 16 rhombi and 8 squares is shown and an algorithm for generating the 8-fold nonperiodic pattern is derived from it. The nonperiodicity is proved by the convergence of the ratio of the number of constituent patterns to irrational value. The diffraction pattern is computed and shown to have spots with 8-fold symmetry. The extension to three dimension or higher order symmetry will be the future problem.

REFERENCES

Bancel,P.A., Heiney,P.A., Stephens,P.W., Goldman,A.I. & Horn,P.M. (1985), Phys. Rev. Lett. **54** 2422-2425

de Bruijn,N.G. (1981), Nederl. Akad. Wetenshch. Proc. Ser. **A43** 39-66

Duneau,M. & Katz,A. (1985), Phys. Rev. Lett. **54** 2688-2691

Elser,V. Acta Crystallogr. **A** (to be published)

Hiraga,K., Hirabayashi,M., Inoue,A. & Masumoto,T. (1985), J. Phys. Soc. Jpn. **54** 4077-4080

Kalugin,P.A., Kitayev,A.Yu. & Levitov,L.S. (1985), J. Physique Lett. **46** L601-L607

Kimura,K., Hashimoto,T., Suzuki,K., Nagayama,K., Ino,H. & Takeuchi,S. (1985), J. Phys. Soc. Jpn. **54** 3217-3219

Kramer,P. & Neri,R. (1984), Acta Crystallogr. **A40** 580-587

Levine,D. & Steinhardt,P.J. (1984), Phys. Rev. Lett. **53** 2477-2480

Mackay,A.L. (1981), Sov. Phys. Crystallogr. **26** 517-522

Mackay,A.L. (1982), Physica **114A** 609-613

Ogawa,T. (1985), J. Phys. Soc. Jpn. **54** 3205-3208

Penrose,R. (1974), Bull. Inst. Maths. Appl. **10** 266-271

Shechtman,D., Blech,I., Gratias,D. & Cahn,J.W. (1984), Phys. Rev. Lett. **53** 1951-1953

(Fig. Aa, Ab, Ac as Appendix 1 are printed on Plate VI at the opening of this volume.)

APPENDIX-2

```
C     DRAW EIGHT-FOLD SYMMETRY PATTERN  SEPT 9.1985 T.SOMA, RIKEN
C
      IMPLICIT REAL*8 (A-H.O-Z)
      COMMON IRET(10).ALENG(10).ANGLE(10),  stack for return adrs
     X ITYPE(10).ISIGN(10).IPAINT(10)       and variables
      DIMENSION IARRAY(5)
      CHARACTER C
C
      PI = ATAN(1.)*4.          define constants
      A18 = PI/8.
      A28 = A18*2.
      A38 = A18*3.
      A48 = A18*4.
      A58 = A18*5.
      A68 = A18*6.
      A78 = A18*7.
      R = 1./(2.*COS(A18))
      S = 1./(2.*COS(A28))
      R1 = COS(A28)/(2.*COS(A28) + 1.)
      R2 = R1/(2.*COS(A18))
      R3 = R2*2.*COS(A28)
      R4 = R2*2.*COS(A38)
      RS = R3/R1
      S3 = R1
      S2 = S3/(2.*COS(A28))
      S1 = S2*2.*COS(A18)
      S4 = S2*2.*COS(A38)
C
    5 WRITE(6.600)             input parameters
  600 FORMAT(' LEN?')
      READ(5.*) PLENG
      WRITE(6.601)
  601 FORMAT(' ANG?')
      READ(5.*) PANGLE
      PANGLE = PANGLE * PI /180.
      WRITE(6.602)
  602 FORMAT(' GEN?')
      READ(5.*) IGEN
      PMINL = PLENG*R1**IGEN + 0.01
      WRITE(6.603)
  603 FORMAT(' ARE YOU SURE?(Y/N)')
      READ(5.500) C
  500 FORMAT(A1)
      IF(C .NE. 'Y'  .AND.  C .NE. ' ') GO TO 5
      CALL PLOTS               initialize plot routine
      I = 1
      ASSIGN 10 TO IW
      CALL PUSH(I.IW.PLENG.PANGLE.1.1.1)
      GO TO 1000               draw rhombus filled
   10 ASSIGN 30 TO IW
      CALL PUSH(I.IW.PLENG.PANGLE.1.1.0)
      GO TO 1000               draw rhombus boundary
   30 CALL PLOTE               finalize plot routine
      STOP
C
C     FUNCTION EIGHT(ALEN.ANG.ITY.ISN.IPA)
 1000 CONTINUE
      CALL POP(I.IW.ALEN.ANG.ITY.ISN.IPA)
      IF(ITY .EQ. 1) THEN
        IF(ALEN .LT. PMINL) THEN      draw rhombus
          IF(IPA .EQ. 1) THEN
            CALL SETARY(1.IARRAY)      set x-y coord
            CALL MOVE(ALEN*R. ANG+A18)     (r-a)
            CALL SETARY(2.IARRAY)
            CALL MOVE(ALEN*R. ANG-A18)     (r-b)
            CALL SETARY(3.IARRAY)
            CALL MOVE(ALEN*R. ANG-A78)     (r-c)
            CALL SETARY(4.IARRAY)
            CALL MOVE(ALEN*R. ANG+A78)     (r-d)
            CALL PATRN('PNT'.'RED')   set fill mode (paint with red)
            CALL POLY(4.IARRAY)       draw polygon
          ELSE
            CALL COLOR('BLK')            set color (black)
            CALL DRAW(ALEN*R. ANG+A18)   draw boundary in black
            CALL DRAW(ALEN*R. ANG-A18)
            CALL DRAW(ALEN*R. ANG-A78)
            CALL DRAW(ALEN*R. ANG+A78)
          ENDIF
          CALL POP(I.IW.ALEN.ANG.ITY.ISN.IPA)
          GO TO IW
        ELSE                          divide rhombus
          I = I + 1
          ASSIGN 1001 TO IW
          CALL PUSH(I.IW.ALEN*R1.ANG.1.0.IPA)        (r-1)
          GO TO 1000
 1001     CALL POP(I-1.IW.ALEN.ANG.ITY.ISN.IPA)
          CALL MOVE(ALEN*(R2+R3), ANG+A18)
          ASSIGN 1002 TO IW
          CALL PUSH(I.IW.ALEN*R3.ANG-A78.0.1.IPA)    (r-2)
          GO TO 1000
 1002     CALL POP(I-1.IW.ALEN.ANG.ITY.ISN.IPA)
          ASSIGN 1003 TO IW
          CALL PUSH(I.IW.ALEN*R1.ANG-A48.1.0.IPA)    (r-3)
          GO TO 1000
 1003     CALL POP(I-1.IW.ALEN.ANG.ITY.ISN.IPA)
          ASSIGN 1004 TO IW
          CALL PUSH(I.IW.ALEN*R3.ANG-A18.0.1.IPA)    (r-4)
          GO TO 1000
 1004     CALL POP(I-1.IW.ALEN.ANG.ITY.ISN.IPA)
          ASSIGN 1005 TO IW
          CALL PUSH(I.IW.ALEN*R3.ANG-A18.0.-1.IPA)   (r-5)
          GO TO 1000
 1005     CALL POP(I-1.IW.ALEN.ANG.ITY.ISN.IPA)
          CALL MOVE(ALEN*R1, ANG)
          ASSIGN 1006 TO IW
          CALL PUSH(I.IW.ALEN*R3.ANG-A18.0.-1.IPA)   (r-6)
          GO TO 1000
 1006     CALL POP(I-1.IW.ALEN.ANG.ITY.ISN.IPA)
          CALL MOVE(ALEN*R2. ANG-A38)
          ASSIGN 1007 TO IW
          CALL PUSH(I.IW.ALEN*R1.ANG+A68. 1.0.IPA)   (r-7)
          GO TO 1000
 1007     CALL POP(I-1.IW.ALEN.ANG.ITY.ISN.IPA)
          ASSIGN 1008 TO IW
          CALL PUSH(I.IW.ALEN*R1.ANG+PI.1.0.IPA)     (r-8)
          GO TO 1000
 1008     CALL POP(I-1.IW.ALEN.ANG.ITY.ISN.IPA)
          ASSIGN 1009 TO IW
          CALL PUSH(I.IW.ALEN*R1.ANG-A68.1.0.IPA)    (r-9)
          GO TO 1000
 1009     CALL POP(I-1.IW.ALEN.ANG.ITY.ISN.IPA)
          ASSIGN 1010 TO IW
          CALL PUSH(I.IW.ALEN*R1.ANG. 1.0.IPA)       (r-10)
          GO TO 1000
 1010     CALL POP(I-1.IW.ALEN.ANG.ITY.ISN.IPA)
          CALL MOVE(ALEN*R2. ANG-A58)
          ASSIGN 1011 TO IW
          CALL PUSH(I.IW.ALEN*R3. ANG+A18.0.1.IPA)   (r-11)
          GO TO 1000
 1011     CALL POP(I-1.IW.ALEN.ANG.ITY.ISN.IPA)
          CALL MOVE(ALEN*R1.ANG+PI)
          ASSIGN 1012 TO IW
          CALL PUSH(I.IW.ALEN*R3.ANG+A18.0.1.IPA)    (r-12)
          GO TO 1000
 1012     CALL POP(I-1.IW.ALEN.ANG.ITY.ISN.IPA)
          ASSIGN 1013 TO IW
          CALL PUSH(I.IW.ALEN*R3.ANG+A18.0.-1.IPA)   (r-13)
          GO TO 1000
 1013     CALL POP(I-1.IW.ALEN.ANG.ITY.ISN.IPA)
          ASSIGN 1014 TO IW
          CALL PUSH(I.IW.ALEN*R3.ANG+A78.0.-1.IPA)   (r-14)
          GO TO 1000
 1014     CALL POP(I-1.IW.ALEN.ANG.ITY.ISN.IPA)
          CALL MOVE(ALEN*(R2+R3).ANG+A78)
          I = I - 1
          CALL POP(I.IW.ALEN.ANG.ITY.ISN.IPA)
          GO TO IW
        ENDIF
      ELSE
        IF(ALEN .LT. PMINL*RS) THEN       draw half square
          IF(IPA .EQ. 1) THEN
            CALL SETARY(1.IARRAY)
            CALL MOVE(ALEN*S.ANG+ISN*A28)        (s-a)
            CALL SETARY(2.IARRAY)
            CALL MOVE(ALEN*S.ANG-ISN*A28)        (s-b)
            CALL SETARY(3.IARRAY)
            CALL MOVE(ALEN. ANG+PI)              (s-c)
            CALL PATRN('PNT'.'GRN')     paint with green
            CALL POLY(3.IARRAY)
          ELSE
            CALL COLOR('BLK')           draw boundary in black
            CALL DRAW(ALEN*S.ANG+ISN*A28)
            CALL DRAW(ALEN*S.ANG-ISN*A28)
            CALL MOVE(ALEN. ANG+PI)
          ENDIF
          CALL POP(I.IW.ALEN.ANG.ITY.ISN.IPA)
          GO TO IW
        ELSE                            divide half square
          I = I + 1
          ASSIGN 1020 TO IW
          CALL PUSH(I.IW.ALEN*S3.ANG.0.ISN. IPA)           (s-1)
          GO TO 1000
 1020     CALL POP(I-1.IW.ALEN.ANG.ITY.ISN.IPA)
          CALL MOVE(ALEN*(S2+S3), ANG+ISN*A28)
          ASSIGN 1021 TO IW
          CALL PUSH(I.IW.ALEN*S3.ANG-ISN*A68.0.ISN.IPA)  (s-2)
          GO TO 1000
 1021     CALL POP(I-1.IW.ALEN.ANG.ITY.ISN.IPA)
          CALL MOVE(ALEN*S1.ANG-ISN*A38)
          ASSIGN 1022 TO IW
          CALL PUSH(I.IW.ALEN*S1.ANG+ISN*A78.1.0.IPA)    (s-3)
          GO TO 1000
 1022     CALL POP(I-1.IW.ALEN.ANG.ITY.ISN.IPA)
          ASSIGN 1023 TO IW
          CALL PUSH(I.IW.ALEN*S1.ANG+ISN*A58.1.0.IPA)    (s-4)
          GO TO 1000
 1023     CALL POP(I-1.IW.ALEN.ANG.ITY.ISN.IPA)
          ASSIGN 1024 TO IW
          CALL PUSH(I.IW.ALEN*S1.ANG+ISN*A38.1.0.IPA)    (s-5)
          GO TO 1000
 1024     CALL POP(I-1.IW.ALEN.ANG.ITY.ISN.IPA)
          ASSIGN 1025 TO IW
          CALL PUSH(I.IW.ALEN*S1.ANG+ISN*A18.1.0.IPA)    (s-6)
          GO TO 1000
 1025     CALL POP(I-1.IW.ALEN.ANG.ITY.ISN.IPA)
          CALL MOVE(ALEN*S1.ANG+ISN*A38)
          ASSIGN 1026 TO IW
          CALL PUSH(I.IW.ALEN*S3.ANG+PI.0.ISN.IPA)       (s-7)
          GO TO 1000
 1026     CALL POP(I-1.IW.ALEN.ANG.ITY.ISN.IPA)
          ASSIGN 1027 TO IW
          CALL PUSH(I.IW.ALEN*S3.ANG+PI.0.-ISN.IPA)      (s-8)
          GO TO 1000
 1027     CALL POP(I-1.IW.ALEN.ANG.ITY.ISN.IPA)
          ASSIGN 1028 TO IW
          CALL PUSH(I.IW.ALEN*S3.ANG-ISN*A28.0.-ISN.IPA) (s-9)
          GO TO 1000
 1028     CALL POP(I-1.IW.ALEN.ANG.ITY.ISN.IPA)
          CALL MOVE(ALEN*(S3+S2).ANG-ISN*A28)
          ASSIGN 1029 TO IW
          CALL PUSH(I.IW.ALEN*S3.ANG+PI.0.-ISN.IPA)      (s-10)
          GO TO 1000
 1029     CALL POP(I-1.IW.ALEN.ANG.ITY.ISN.IPA)
          CALL MOVE(ALEN*(2.*S3+2.*S2), ANG+PI)
          I = I - 1
          CALL POP(I.IW.ALEN.ANG.ITY.ISN.IPA)
          GO TO IW
        ENDIF
      ENDIF
      END
C
      SUBROUTINE PUSH(I.IW.ALEN.ANG.ITY.ISN.IPA)    push stack
      IMPLICIT REAL*8 (A-H.O-Z)
      COMMON IRET(10).ALENG(10).ANGLE(10),
     X ITYPE(10).ISIGN(10).IPAINT(10)
        IRET(I) = IW
        ALENG(I) = ALEN
        ANGLE(I) = ANG
        ITYPE(I) = ITY
        ISIGN(I) = ISN
```

```
      IPAINT(I) = IPA
      RETURN
      END
c
      SUBROUTINE POP(I,IW,ALEN,ANG,ITY,ISN,IPA)        pop stack
      IMPLICIT REAL*8 (A-H,O-Z)
      COMMON IRET(10),ALENG(10),ANGLE(10),
     X ITYPE(10),ISIGN(10),IPAINT(10)
      IW = IRET(I)
      ALEN = ALENG(I)
      ANG = ANGLE(I)
      ITY = ITYPE(I)
      ISN = ISIGN(I)
      IPA = IPAINT(I)
      RETURN
      END
c
      SUBROUTINE SETARY(I,IARRAY)       set array of coord
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION IARRAY(1)
      CALL WHEREA(XX,YY)                get absolute coord
      IARRAY(2*I-1) = XX + 0.51         round-off to integer
      IARRAY(2*I)   = YY + 0.51
      RETURN
      END
c
      SUBROUTINE MOVE(R,T)              move and shift origin
      IMPLICIT REAL*8 (A-H,O-Z)
      CALL PLOT(R*COS(T), R*SIN(T), -3)
      RETURN
      END
c
      SUBROUTINE DRAW(R,T)              draw and shift origin
      IMPLICIT REAL*8 (A-H,O-Z)
      CALL PLOT(R*COS(T), R*SIN(T), -2)
      RETURN
      END
```

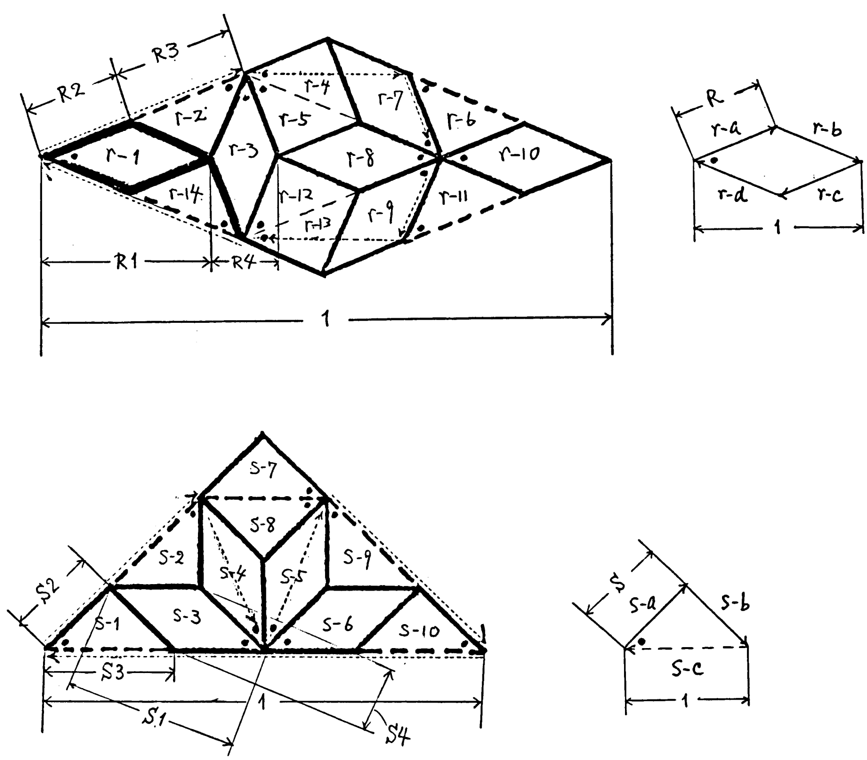

Fig. A d   Drawing path