match no. 20 pp. 251-279 1986

CONTEXT-FREE GRAMMAR AND DETERMINISTIC AUTOMATON APPROACHES FOR SEQUENCE GENERATION IN COPOLYMERS. I. BINARY COPOLYMERS WITH ULTIMATE EFFECT

R. Vancea[a)], St. Holban[b)] and D. Ciubotariu[b)]

a) Computing Center
5800 SUCEAVA
ROMANIA

b) Polytechnic Institute "Traian Vuia"
1900 TIMISOARA
ROMANIA

(Received: May 1986)

1. Abstract.

Linguistic pattern recognition techniques have been used for generating and identifying the near sequence in binary copolymers on the basis of the ultimate effect.

2. Introductory Notions in Linguistic Pattern Recognition.

Linguistic recognition uses the analogy between letters and the words within a phrase and the structure of a pattern. In this analogy any pattern (geometrical feature, digital print, image, etc.) [1, 2] may be decomposed into a number of component parts (called primitives) that have for correspondent the letters and the words of a language. On the other side the rules of a pattern composition may be related to the grammar rules of building a phrase. Therefore we come to the conclusion that defining a language and a grammar that describes a certain pattern class is possible. In order to exemplify let us consider the geometrical pattern of a square. The square can be decomposed (Figure 1) into a number of four primitives a, b, c, d that define unit length segments. Therefore any sequence of four letters of abcd, bcda, cdab, dabc type will identify the

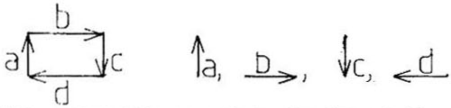


FIGURE 1. Primitives used in the description of a square.

square. Further on, the language L_1 that generates the set of the squares having sides of length equal to n units will be defined by the set:

$$L_1 = \left\{ a^n\ b^n\ c^n\ d^n \ \middle|\ n \geqslant 1 \right\}$$

In order to specify the rules of building squares using elements of this language we use the theory of pattern languages that defines a grammar of the following pattern:

$$G = (V_N, V_T, P, S)$$

where:

V_N - the set of the non-terminal symbols;

V_T - the set of the terminal symbols;

P - grammar rules;

S - initial symbol.

The grammar elements that generate tha patterns described by means of the L_1 language are:

$$V_T = \left\{ a, b, c, d \right\}$$

$$V_N = \left\{ S, A, B, C, D \right\}$$

$$\begin{array}{llll} P : S \dashrightarrow a\,A, & A \dashrightarrow a\,A, & B \dashrightarrow b\,B, & D \dashrightarrow d\,D \\ & A \dashrightarrow B & B \dashrightarrow c\,C & D \dashrightarrow d \\ & A \dashrightarrow b\,B & C \dashrightarrow d\,D & \end{array}$$

The operation through which we build the pattern, following the grammar rules specified by P is called derivation and is represented by the symbol "$\Longrightarrow$". In the case of a square having unit sides, the derivation may have the pattern:

$$S \Longrightarrow a\,A \Longrightarrow a\,b\,B \Longrightarrow a\,b\,c\,C \Longrightarrow a\,b\,c\,D \Longrightarrow a\,b\,c\,d$$

Such a grammar is known under the name of context-free grammar. If we wish to represent a quadrilateral, whose primitives are described in Figure 2, the language will have the pattern:

$$L_2 = \left\{ a^i b^j c^k d^l \;\middle|\; i,j,k,l \geqslant 1 \right\}$$

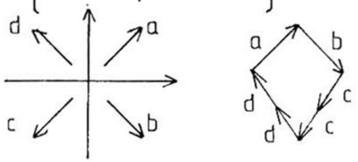


FIGURE 2. Primitives used in the description of a quadrilateral.

In this case the grammar is stochastic and has the pattern:

$$G_S = (V_N, V_T, P_S, S)$$

where P_S is a finite set of stochastic rules.

For a context-free stochastic grammar a rule in P_S has the pattern:

$$A_i \xrightarrow{P_{ij}} \alpha_i, \; A_i \in V_N, \alpha_i \in (V_N \cup V_T)$$

where P_{ij} represents the production probability.

In the case of a quadrilateral the grammar rules are given by:

$$P_S\!: \; S \dashrightarrow aA, \; A \xrightarrow{P_{11}} aA, \; B \xrightarrow{P_{22}} bB, \; C \xrightarrow{P_{33}} cC, \; D \xrightarrow{P_{44}} dD$$

$$A \xrightarrow{P_{12}} B \quad B \xrightarrow{P_{23}} C \quad C \xrightarrow{P_{34}} D \quad D \dashrightarrow d$$

Supposing the probability distribution is uniform within the interval (0, 1), the derivation rules for a quadrilateral are:

$$S \Longrightarrow aA \xRightarrow{P_{12}} aB \xRightarrow{P_{22}} abB \xRightarrow{P_{23}} abC \xRightarrow{P_{33}} abcC \xRightarrow{P_{33}}$$

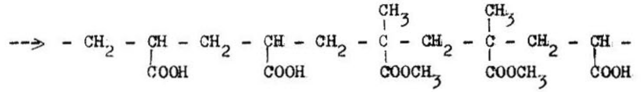
$$\xRightarrow{P_{33}} abccC \xRightarrow{P_{34}} abccD \xRightarrow{P_{44}} abccdD$$

3. Generating Copolymer Sequences.

A copolymer represents a chemical macromolecular compound having an aleatory repetitive structure in which the setting-up of mer sequences (the mer being a structural elementary unit of a macromolecular chain) as well as their nature and number depend on the conditions in which the chemical reaction of copolymer formation takes place (temperature, pressure, solvent catalyst, etc.). The nature and the reactivity of the monomers also influence the mer sequence. As there is a close link between the copolymer structure and its properties, any theoretical approach in determining the structure on the ground of the monomer reactivity may prove to be extremely useful as it might allow the synthesis of macromolecular compounds having an a-priori structure. Therefore, determining sequence distribution is important for understanding physical properties of the copolymer, for determining reactivity parameters of monomers and for discriminating between many possible reaction mechanisms. In order to exemplify we present a binary copolymerization reaction between acrylic acid (M_1) and methyl methacrylate (M_2) monomers, as well as a possible $M_1M_1M_2M_2M_1$ sequence in the copolymer chain:

$$CH_2 = CH - COOH + CH_2 = C(CH_3) - COOCH_3 \dashrightarrow$$

$$(M_1) \qquad\qquad\qquad (M_2)$$



Several models have been developed to explain the way in which copolymerization reactions take place and to allow copolymer structure determination according to monomer reactivity { 4 }. All these studies take into account the appearance probability distribution of a mer conditioned by the growing macro-radical structure (the ultimate mer-model and the ultimate effect, the penultimate mer-models and penultimate effect etc.) as well as the conditions of setting up a copolymerization reaction, expressed through r_1 and r_2 copolymerization reactivity ratios. A binary copolymerization reaction (two monomers) with an ultimate effect (that is only terminal monomer units - mers - alter the rate values) imply four elementary growing reactions [3]:

$$\sim M_I^* + M_J \xrightarrow{K_{IJ}} \sim M_J^*, \quad I, J = 1,2$$

The ratios $r_I = K_{II} / K_{IJ}$, I, J = 1,2, I ≠ J are called reactivity ratios. Supposing that:

i) rate values do not depend on the length of the growing macroradical;

ii) we can neglect initiation or ending processes (that is we consider that the polymer length is so large that it does not influence initiation or ending macroradical growing), the probabilities of transition P_{II} from the state M_I^* to the state M_J^* can easily be calculated thus:

$$P_{II} = \frac{K_{II}(M_I^*)(M_I)}{K_{II}(M_I^*)(M_I) + K_{IJ}(M_I^*)(M_J)} =$$

$$= \frac{K_{II}(M_I)}{K_{II}(M_I) + K_{IJ}(M_J)} = \frac{r_I (M_I)}{r_I (M_I) + (M_J)} = \frac{r_I}{r_I + C}$$

where ( ) designs the initial monomer concentration and $C = (M_J) / (M_I)$. P_{IJ} can be obtained with the following relation:

$$P_{IJ} = 1 - P_{II} = \frac{r_I}{r_I + C}$$

The authors intend to develop a linguistic copolymer sequence generation and recognition system. They have thus proposed the interpretation of a copolymer sequence as sentences of different lengths in which the mers are letters that group in words formed of one or many mers of the same type. Here are the steps we have made:

1° Choosing the primitives is imposed by the mer type that composes the copolymer. As in the case of binary copolymerization the monomer units are of two different types, that have been chosen as primitives, the mers (the primitives) being noted with M_1 and M_2. As noted above, in the case of the copolymer acrylic acid / methyl methacrylate, primitive M_1 will be represented by the mer $- CH_2 - C(CH_3)(COOCH_3) -$.

2° Copolymer structure representation is made under the form of a primitive string. Thus, for the situation mentioned above a sequence in a primitive string is .... $M_1M_1M_2M_1M_2M_2$...

3° A context-free stochastic grammar of the pattern below has been chosen for describing and generating sentences that represent such structures:

$$G_S = (V_N, V_T, P_S, S)$$

where:

$$V_N = \{S, X, Y\}$$

$$V_T = \{M_1, M_2\}$$

$$P_S : S \dashrightarrow M_1X, \quad X \xrightarrow{P_{11}} M_1X, \quad Y \xrightarrow{P_{21}} M_1X$$

$$S \dashrightarrow M_2Y \quad X \xrightarrow{P_{12}} M_2Y \quad Y \xrightarrow{P_{22}} M_2Y$$

$$P_{11} = \frac{r_1}{r_1 + C}, \quad P_{12} = 1 - P_{11}, \quad P_{22} = \frac{r_2}{r_2 + C}$$

$P_{21} = 1 - P_{22}$; r_1 and r_2 are copolymerization rate values.

The grammar thus defined has been implemented in a program for generating copolymer sequence in which the copolymerization degree represents the sequence building process stepping criterion [4 - 11]. The probability of generating the string that represents the copolymer appears as the product of all probability productions associated with the production used in mer generating. We have considered that if the last mer is of the M_1 or M_2 type the appearance probability of another M_1 or M_2 mer to be attached to the growing chain is uniformely distributed; the appearance probabilities are P_{11}, P_{12} respectively if the last mer is of the M_2 type. A sequence generated according to this grammar is shown in Figure 3, and the GENERATE-STRING sequence generation program is presented in the Appendix 1.

$M_1-M_2-M_1-M_1-M_2-M_1-M_2-M_2-M_1-M_2-M_2-M_1-M_1-M_2-M_1-M_2-M_2-M_2-M_1-M_2-M_1-M_1-M_1-M_1-$

$M_1-M_1-M_2-M_1-M_1-M_1-M_2-M_1-M_2-M_1-M_1-M_1-M_2-M_2-M_2-M_2-M_2-M_2-M_2-M_2-M_1-M_1-M_1-M_2-$

$M_1-M_1-M_2-M_1-M_2-M_2-M_1-M_2-M_2-M_2-M_1-M_2-M_1-M_2-M_1-M_2-M_1-M_1-M_1-M_1-M_2-M_1-M_1-M_1-$

$M_1-M_2-M_2-M_1-M_1-M_2-M_1-M_2-M_1-M_1-M_1-M_1-M_1-M_1-M_1-M_1-M_1-M_2-M_1-M_1-M_1-M_1-M_1-M_1-$

$M_1-M_1-M_1-M_1-M_1-M_1-M_1-M_1-M_1-M_2-M_2-M_1-M_1-M_1-M_1-M_1-M_1-M_1-M_1-M_1-M_1-M_2-M_1-M_1-$

$M_1-M_2-M_1-M_2-M_2-M_2-M_2-M_1-M_1-M_2-M_1-M_1-M_1-M_2-M_2-M_1-M_1-M_1-M_1-M_1-M_2-M_1-M_1-M_1-$

....

....

$M_2-M_1-M_2-M_1-M_2-M_1-M_1-M_2-M_1-M_1-M_1-M_1-M_1-M_1-M_1-M_1-M_1-M_2-M_2-M_1-M_2-M_1-M_2-M_1-$

$M_2-M_1-M_1-M_1-M_1-M_1-M_1-M_2-M_1-M_1-M_2-M_1-M_1-M_1-M_2-M_1-M_2-M_2-M_1-M_1-M_2-M_1-M_2-M_1$

FIGURE 3. (M_1*M_2) 2-vinyl-pyridine*styrene copolymer partial string sequence generated according to G_S.

4. Principles of Copolymer Linguistic Recognition.

If a pattern class can be described through a determinist language then the procedure of recognizing a pattern to belong to a certain class is made by a determinist automaton. This accepts as the input the string describing the pattern, and verifies the belonging of a pattern to a certain class with the help of grammar rules used in building a pattern. For the pattern generated by the L language $L = \left\{ a^n b^n c^n d^n \mid n \geq 1 \right\}$ the determinist automaton has the pattern:

$$A = (\Sigma, Q, \ , q_0, F)$$

where:

$\Sigma = \left\{ a, b, c, d \right\}$ and is the alphabet;

$Q = \left\{ q_0, q_1, q_2, q_3 \right\}$ the state set;

$F = \left\{ q_3 \right\}$ final state set;

q_0 is the initial state.

δ is an application on the Q x F set with values in the Q set.

The transition diagram has the pattern:

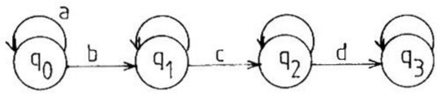


FIGURE 4. The transition diagram of the A automaton.

For the A automaton, will have the values:

$$\delta(q_0, a) = \left\{ q_0, q_1 \right\}$$

$$\delta(q_1, b) = \left\{ q_1, q_2 \right\}$$

$$\delta(q_2, c) = \left\{ q_2, q_3 \right\}$$

$$\delta(q_3, d) = \left\{ q_3 \right\}$$

In the case of pattern recognition whose generation was made with the help of a stochastic grammar, the most simple procedure of recognition can be made with the help of a non-determinist automaton. This solution has been adopted by the authors for the recognition (identification) of a copolymer. The next non-determinist automaton has been defined:

$C = (\Sigma, Q, q_0, F)$

where:

$\Sigma = \{M_1, M_2\}$

$Q = \{q_0, q_1, q_2\}$

$F = \emptyset$

$\delta(q_0, M_1) = \{q_1, q_2\}$

$\delta(q_1, M_1) = \{q_1\}$

$\delta(q_2, M_2) = \{q_2\}$

$\delta(q_2, M_1) = \{q_1\}$

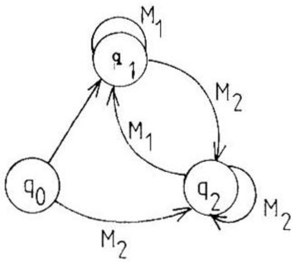


FIGURE 5. The transition diagram of the C automaton.

The transition diagram of the C automaton is given in Figure 5. The automaton was implemented through a computer program. This program accepts as input the string that describes the copolymer, and affords the rate values as output. We have calculated the rate values r_1 and r_2 of the copolymer that results from the methyl methacrylate / chloroprene copolymerization:

$r_1 = 0.073$ and $r_2 = 5.852$

(the experimental [12] ones having the following values: $r_1 = 0.08$ and $r_2 = 5.1$). The RECOGNITION-STRING program is presented in the Appendix 2.

5. Results.

In order to exemplify (See Table 1) we offer below the results obtained in the case of copolymerization of acrylic acid (M_1) with methyl methacrylate (M_2).

	M_1		M_1 in the obtained copolymer	
	initial	experimental	Markov (M)	Automaton (C)
1.	88.0	91.1	90.8	92.0
2.	88.1	85.7	84.9	85.4
3.	72.4	77.2	77.2	76.7
4.	71.6	75.6	76.4	75.5
5.	58.9	61.5	63.9	61.2
6.	58.7	62.2	63.7	61.1
7.	51.0	56.1	55.6	53.4
8.	45.8	49.1	50.0	46.6
9.	34.0	41.1	36.9	33.0
10.	30.5	32.6	33.0	28.7
11.	20.2	25.1	21.5	20.2

TABLE 1.

The agreement between the compositions calculated using a Markov chain model of the first order and the stochastic linguistic grammar described above is very good, as it is shown in the following linear regression equations:

$$\% M_1, \exp = 7.658 + 0.904 \quad \% M_1, C$$

$$(r = 0.998, s = 1.371, F = 981.779)$$

$$\% M_1, \exp = 2.854 + 0.957 \quad \% M_1, M$$

$$(r = 0.99, s = 1.759, F = 594,962)$$

where r represents the correlation coefficient, s the standard deviation and F the Fisher statistics.

Acknowledgement.

We wish to thank to Professor A.T. Balaban for the most useful advice and suggestions he gave us both throughout the development of the GENERATE-STRING and RECOGNITION-STRING programs as well as in writing

this paper.

Bibliography.

1. K.S. Fu, Sequential Methods in Pattern Recognition and Machine Learning, Academic Press, New York, 1968.

2. K.S. Fu ed., Digital Pattern Recognition, Springer-Verlag, Berlin, 1980.

3. M. Izu and K.F. O'Driscoll, J. Polym. Sci., A - 1,8, 1675, 1970.

4. W. Bruns, I. Motoc, K.F. O'Driscol, Monte Carlo Applications in Polymer Science, Lecture Notes in Chemistry, No. 27, Springer, 1981.

5. I. Motoc, St. Holban and D. Ciubotariu, J. Polym. Sci., 15, 1465, 1977.

6. I. Motoc, D. Ciubotariu and St. Holban, Rev. Roumaine Chim., 21, 775, 1976.

7. I. Motoc, St. Holban and R. Vancea, J. Polym. Sci., 16, 1601, 1978.

8. I. Motoc, R. Vancea and St. Holban, J. Polym. Sci., 16, 1587 - 1593, 1978.

9. I. Motoc, R. Vancea and St. Holban, J. Polym. Sci., 16, 1595 - 1599, 1978.

10. I. Motoc and R. Vancea, J. Polym. Sci., 18, 1559 - 1564, 1980.

11. I. Motoc, R. Vancea and I. Muschiutariu, Monte Carlo Applications in Chemical Physics of Polymer - The Memory Program, Reprint, Timişoara University, 1980.

12. D. Braun, W. Brendlein and G. Mott, Eur. Polym., J., 9, 1007 (1973).

APENDIX 1

```
C *********************************************************************
C *******                                                           ***
C *******                                                           ***
C *******           G E NE R A T E - S T R I N G                    ***
C *******                                                           ***
C *******                                                           ***
C *********************************************************************

C         GENERATE-STRING IS NEW VERSION OF MEMORY-3


      BLOCK DATA

      COMMON/TMER/PKTA,PKTB,BLK

      DATA PKTA,PKTB,BLK/2HA-,2HB-,1H /

      END

      INTEGER SW,FRMOL,MER,NA,NB,NTAS,NTBS,NRA,NRB,NBA,NBB,RA,RB

      COMMON N

      COMMON/KONST/SW,FRMOL,MER,NA,NB,NTAS,NTBS,NRA,NRB,NBA,NBB,RA,RB,I

      COMMON/POLYM/POLYM(5000)/DIVID/ALPHA,BETA,C,R1,R2

      COMMON/TABEL/GTAB(5,16),J/RANDOM/IA,IB,X

      DIMENSION NAME(40)

      L=108

1     READ(105,900,END=8)(NAME(K),K=1,40)

      READ(105,910)MER,N,R1,R2,NA,NB,FRMOL

      WRITE(L,940)

      WRITE(L,900)(NAME(K),K=1,40)

      WRITE(L,950)N,R1,R2,FRMOL,NA,NB

      CALL INIT

      CALL ERROR(MC)

      IF(MC.EQ.1)GO TO 1

      IF(MER.EQ.1)CALL INITA

      IF(MER.EQ.2)CALL INITB

2     CALL PART
```

```
3       CALL ERRORP(MC)
        IF(MC.EQ.1)GO TO 1
        IF(MC.EQ.1)GO TO 1
4       CALL MAS
        CALL ALEAT
        CALL NMAS
        IA=IB
        CALL ERRORR(MC)
        IF(MC.EQ.1)GO TO 1
        I=I+1
        IF(I.GT.N)CALL OUT(MC)
        IF(MC.EQ.1)GO TO 1
        GO TO (5,6),SW
5       CALL GENER1
        GO TO 7
6       CALL GENER2
7       CALL REEW(MC)
        IF(MC)2,3,4
8       STOP
900     FORMAT(40A2)
910     FORMAT(I2,I6,2F14.8,3I6)
940     FORMAT(1H0////)
950     FORMAT(5X,'POLYMERISATION DEGREE=',I6/
     1        5X,'                      R1=',F14.8/
     2        5X,'                      R2=',F14.8/
     3        5X,'EVALUTION MODE =',I6/
     4        5X,'                      NA=',I6/
     4        5X,'                      NB=',I6//)
        END
        SUBROUTINE REEW(MC)
```

```
      INTEGER SW,FRMOL,MER,NA,NB,NTAS,NTBS,NRA,NRB,NBA,NBB,RA,RB
      COMMON/KONST/SW,FRMOL,MER,NA,NB,NTAS,NTBS,NRA,NRB,NBA,NBB,RA,RB,I
      COMMON/POLYM/POLYM(5000)/DIVID/ALPHA,BETA,C,R1,R2
      COMMON/TMER/PKTA,PKTB,BLK
      IF(((ALPHA.EQ.0.).AND.(BETA.EQ.0.)).OR.
     1   ((ALPHA.EQ.1.).AND.(BETA.EQ.1.))) GO TO 1
      IF((NTAS+NTBS).LT.FRMOL) GO TO 1
      NA=NA-NTAS
      NB=NB-NTBS
      NTAS=0
      NTBS=0
      IF((NB.LT.0).OR.(NA.LT.0)) GO TO 2
      IF(NA.EQ.0) GO TO 3
      IF(NB.EQ.0) GO TO 4
      MC=-1
      RETURN
2     MC=0
      RETURN
3     CALL ENDA
1     MC=1
      RETURN
4     CALL ENDB
      MC=1
      RETURN
      END
      SUBROUTINE INIT
      INTEGER SW,FRMOL,MER,NA,NB,NTAS,NTBS,NRA,NRB,NBA,NBB,RA,RB
      COMMON/KONST/SW,FRMOL,MER,NA,NB,NTAS,NTBS,NRA,NRB,NBA,NBB,RA,RB,I
      COMMON/POLYM/POLYM(5000)/DIVID/ALPHA,BETA,C,R1,R2
      COMMON/TABEL/GTAB(5,16),J/RANDOM/IA,IB,X
```

```
      COMMON/TMER/PKTA,PKTB,BLK
      I=1
      J=1
      IA=65539
      NRA=0
      NRB=0
      NBA=0
      NBB=0
      NTAS=0
      NTBS=0
      DO 1 L=1,5
      DO 1 K=1,16
1     GTAB(L,K)=BLK
      DO 2 L=1,20
      CALL MAS
      CALL ALEAT
      CALL NMAS
2     IA=IB
      RETURN
      ENTRY INITA
      POLYM(1)=PKTA
      SW=1
      NRA=1
      NTAS=1
      RETURN
      ENTRY INITB
      POLYM(1)=PKTB
      SW=2
      NRB=1
      NTBS=1
```

```
      RETURN
      END
      SUBROUTINE PART
      INTEGER SW,FRMOL,MER,NA,NB,NTAS,NTBS,NRA,NRB,NBA,NBB,RA,RB
      COMMON/KONST/SW,FRMOL,MER,NA,NB,NTAS,NTBS,NRA,NRB,NBA,NBB,RA,RB,I
      COMMON/POLYM/POLYM(5000)/DIVID/ALPHA,BETA,C,R1,R2
      C=FLOAT(NA)/FLOAT(NB)
      ALPHA=R1/(1./C+R1)
      BETA=1.-R2/(C+R2)
      RETURN
      END
      SUBROUTINE ERROR(MC)
      INTEGER SW,FRMOL,MER,NA,NB,NTAS,NTBS,NRA,NRB,NBA,NBB,RA,RB
      COMMON N
      COMMON/KONST/SW,FRMOL,MER,NA,NB,NTAS,NTBS,NRA,NRB,NBA,NBB,RA,RB,I
      COMMON/POLYM/POLYM(5000)/DIVID/ALPHA,BETA,C,R1,R2/RANDOM/IA,IB,X
      MC=0
      L=108
      IF(N.LE.0) GO TO 1
      IF(NB.LE.0.OR.NA.LE.0) GO TO 2
      IF(FRMOL.LE.0) GO TO 3
      IF(MER.LT.1.OR.MER.GT.2) GO TO 4
      RETURN
1     WRITE(L,900)N
      MC=1
      RETURN
2     WRITE(L,910)NA,NB
      MC=1
      RETURN
3     WRITE(L,920)FRMOL
```

```
      MC=1
      RETURN
4     WRITE(L,930)MER
      MC=1
      RETURN
900   FORMAT(///5X,'*ERORR* N=',I6)
910   FORMAT(///5X,'*ERORR* NA=',I6,' NB=',I6)
920   FORMAT(///5X,'*ERORR* EVALUATION MODE=',I6)
930   FORMAT(///5X,'*ERORR* MER=',I2)
      ENTRY ERRORP(MC)
      L=108
      MC=0
      IF((ALPHA.LT.0.).OR.(ALPHA.GT.1.).OR.
     1(BETA.LT.0.).OR.(BETA.GT.1)) GO TO 5
      RETURN
5     WRITE(L,940)ALPHA,BETA
      MC=1
      RETURN
940   FORMAT(///5X,'*ERORR* ALPHA=',F14.8,' BETA=',F14.8)
      ENTRY ERRORR(MC)
      MC=0
      L=108
      IF(X.LE.0.OR.X.GE.1.) GO TO 6
      RETURN
6     WRITE(L,960)X
      MC=1
      RETURN
960   FORMAT(///5X,'*ERORR* GENERATOR DEFECT X=',F14.7)
      END
      SUBROUTINE GENER1
```

```
      INTEGER SW,FORMOL,MER,NA,NB,NTAS,NTBS,NRA,NRB,NBA,NBB,RA,RB
      COMMON/KONST/SW,FRMOL,MER,NA,NB,NTAS,NTBS,NRA,NRB,NBA,EBB,RA,RB,I
      COMMON/POLYM/POLYM(5000)/DIVID/ALPHA,BETA,C,R1,R2
      COMMON/TABEL/GTAB(5,16),J/RANDOM/IA,IB,X
      COMMON/TMER/PKTA,PKTB,BLK
1     IF(NA.GT.NTAS) GO TO 2
      IF(NA.EQ.0.AND.NTAS.EQ.0) GO TO 2
      CALL ENDA
      GO TO 3
2     IF(X.LE.ALPHA) GO TO 5
3     IF(NRA.EQ.1.OR.NRA.EQ.0) GO TO 4
      RA=NA-NTAS
      RB=NB-NTBS
      NBA=NBA+1
      GTAB(1,J)=PKTA
      GTAB(2,J)=NRA
      GTAB(3,J)=NBA
      GTAB(4,J)=RA
      GTAB(5,J)=RB
      J=J+1
      IF(J.GT.14) CALL GRTAB
4     POLYM(I)=PKTB
      SW=2
      NRA=0
      NRB=NRB+1
      NTBS=NTBS+1
      RETURN
5     POLYM(I)=PKTA
      NRA=NRA+1
      NTAS=NTAS+1
```

```
      RETURN
      END
      SUBROUTINE GENER2
      INTEGER SW,FRMOL,MER,NA,NB,NTAS,NTBS,NRA,NRB,NBA,NBB,RA,RB
      COMMON/KONST/SW,FRMOL,MER,NA,NB,NTAS,NTBS,NRA,NRB,NBA,NBB,RA,RB,I
      COMMON/POLYM/POLYM(5000)/DIVID/ALPHA,BETA,C,R1,R2
      COMMON/TABEL/GTAB(5,16),J/RANDOM/IA,IB,X
      COMMON/TMER/PKTA,PKTB,BLK
1     IF(NB.GT.NTBS) GO TO 2
      IF(NB.EQ.0.AND.NTBS.EQ.0) GO TO 2
      CALL ENDB
      GO TO 3
2     IF(X.LE.BETA)  GO TO 3
      POLYM(I)=PKTB
      NRB=NRB+1
      NTBS=NTBS+1
      RETURN
3     IF(NRB.EQ.1.OR.NRB.EQ.0) GO TO 4
      RA=NA-NTAS
      RB=NB-NTBS
      NBB=NBB+1
      GTAB(1,J)=PKTB
      GTAB(2,J)=NRB
      GTAB(3,J)=NBB
      GTAB(4,J)=RA
      GTAB(5,J)=RB
      J=J+1
      IF(J.GT.14) CALL GRTAB
4     POLYM(I)=PKTA
      SW=1
```

```
      NRB=0
      NRA=NRA+1
      NTAS=NTAS+1
      RETURN
      END
      SUBROUTINE  OUT(MC)
      INTEGER SW,FRMOL,MER,NA,NB,NTAS,NTBS,NRA,NRB,NBA,NBB,RA,RB
      INTEGER G(2,40),A,B
      COMMON N
      COMMON/KONST/SW,FRMOL,MER,NA,NB,NTAS,NTBS,NRA,NRB,NBA,NBB,RA,RB
      COMMON/POLYM/POLYM(5000)/DIVID/ALPHA,BETA,C,R1,R2
      COMMON/TABEL/GTAB(5,16),J/RANDOM/IA,IB,X
      COMMON/TMER/PKTA,PKTB,BLK
      MC=0
      L=108
      IF(NRA.EQ.1.OR.NRA.EQ.0) GO TO 1
      RA=NA-NTAS
      RB=NB-NTBS
      NBA=NBA+1
      GTAB(1,J)=PKTA
      GTAB(2,J)=NRA
      GTAB(3,J)=NBA
      GTAB(4,J)=RA
      GTAB(5,J)=RB
1     IF(NRB.EQ.1.OR.NRB.EQ.0) GO TO 2
      RA=NA-NTAS
      RB=NB-NTBS
      NBB=NBB+1
      GTAB(1,J)=PKTB
      GTAB(2,J)=NRB
```

```
      GTAB(3,J)=NBB
      GTAB(4,J)=RA
      GTAB(5,J)=RB
2     IF(J.GT.1) CALL GRTAB
      IF(J.EQ.1.AND.(GTAB(1,J).EQ.PKTA.OR.GTAB(1,J).EQ.PKTB))CALL GRTAB
      DO 3 I=1,2
      DO 3 J=1,40
3     G(I,J)=0
      NJ=0
      A=0
      B=0
      IC=-1
      IF(POLYM(1).EQ.PKTA)IC=1
      DO 20 I=1,N
      IF(POLYM(I).EQ.PKTA) GO TO 4
      GO TO 8
4     IF(IC.EQ.-1)GO TO 5
      A=A+1
      GO TO 20
5     IF(B.GT.40) GO TO 7
      G(2,B)=G(2,B)+1
6     IC=1
      B=0
      GO TO 4
7     G(2,40)=G(2,40)+1
      GO TO 6
8     IF(IC.EQ.1) GO TO 9
      B=B+1
      GO TO 20
9     IF(A.GT.40) GO TO 11
```

```
      G(1,A)=G(1,A)+1
10    IC=-1
      A=0
      NJ=NJ+1
      GO TO 8
11    G(1,40)=G(1,40)+1
      GO TO 10
20    CONTINUE
      IF(IC.EQ.-1) GO TO 30
      IF(A.GT.40) A=40
      G(1,A)=G(1,A)+1
      GO TO 40
30    IF(B.GT.40) B=40
      G(1,B)=G(1,B)+1
40    CONTINUE
      WRITE(L,920)(J,J=1,40),((G(I,J),J=1,40),I=1,2)
920   FORMAT(//5X,'CONFIGURATION'/5X,12('-')/
     14X,'N I',40I3/4X,123('-')/4X,'A I',40I3/4X,'B I',40I3)
C     WRITE(6,900)(POLYM(I),I=1,N)
      WRITE(L,910)NBA,NBB,NJ
900   FORMAT(//5X,'THE MOST PROBABLE SHAPE OF THE MOLECULE'
     1          /5X,'----------------------------------------',
     2         //(6X,30A2))
910   FORMAT(/5X,'NUMBER OF GROUPS A=',I7
     1         /5X,'NUMBER OF GROUPS B=',I7
     1/5X,'NUMBER OF GROUPS A-B =',I7)
      MC=1
      RETURN
      END
      SUBROUTINE GRTAB
```

```
      COMMON/TABEL/GTAB(5,16),J/RANDOM/IA,IB,X
      COMMON/POLYM/POLYM(5000)/DIVID/ALPHA,BETA,C,R1,R2
      COMMON/TMER/PKTA,PKTB,BLK
      L=108
C     WRITE(6,900)((GTAB(II,JJ),JJ=1,14),II=1,5)
      DO 1 II=1,5
      DO 1 JJ=1,16
1     GTAB(II,JJ)=BLK
      J=1
      RETURN
900   FORMAT(//2X,130('-')/2X,'! TYPE        !',
     1             14('I   ',A2,'   '),' !',
     2        /2X,130('-')/2X,'! LENGTHS          !',
     3             14('! ',F5.0),' !',
     4        /2X,130('-')/2X,'! POZITION         !',
     5             14('! ',F5.0),' !',
     6        /2X,130('-')/2X,'! NUMBER OF MERS A IN FGED  !',
     7             14('! ',F5.0),' !',
     8        /2X,130('-')/2X,'! NUMBER OF MERS B IN FEED  !',
     9             14('! ',F5.0),' !',
     9        /2X,130('-'))
      END
      SUBROUTINE ENDB
      INTEGER SW,FRMOL,MER,NA,NB,NTAS,NTBS,NRA,NRB,NBA,NBB,RA,RB
      COMMON/KONST/SW,FRMOL,MER,NA,NB,NTAS,NTBS,NRA,NRB,NBA,NBB,RA,RB,I
      COMMON/TABEL/GTAB(5,16),J/RANDOM/IA,IB,X
      COMMON/POLYM/POLYM(5000)/DIVID/ALPHA,BETA,C,R1,R2
      L=108
      ALPHA=1.0
      BETA=1.0
```

```
      NTBS=0
      FRMOL=2+N
      NB=0
      WRITE(L,900)
900   FORMAT(5X,'*** NUMBER OF MERS B = 0     *')
      RETURN
      END
      SUBROUTINE ENDA
      INTEGER SW,FRMOL,MER,NA,NB,NTAS,NTBS,NRA,NRB,NBA,NBB,RA,RB
      COMMON/TABEL/GTAB(5,16),J/RANDOM/IA,IB,X
      COMMON/KONST/SW,FRMOL,MER,NA,NB,NTAS,NTBS,NRA,NRB,NBA,NBB,RA,RB,I
      COMMON/POLYM/POLYM(5000)/DIVID/ALPHA,BETA,C,R1,R2
      L=108
      ALPHA=0.0
      BETA=0.0
      NTAS=0
      FRMOL=2+N
      NA=0
      WRITE(L,900)
900   FORMAT(5X,'*** NUMBER OF MERS A = 0 *** ')
      RETURN
      END
      SUBROUTINE ALEAT
      COMMON/RANDOM/IA,IB,X
      CALL RAN
      RETURN
      END
      SUBROUTINE RAN
      COMMON/RANDOM/IA,IB,X
      IB=IA*65539
```

```
      IF(IB.GE.0) GO TO 1
      IB=IB+2147483647+1
    1 X=IB
      X=X*0.456613E-09
      IA=IB
      RETURN
      END
.           ENTSG MAS,NMAS
.           COMPILE ASSIRIS
MASKEI      CSECT        P
            DEF          MAS,NMAS
MASQUE      DATA,4,4     X'50400000'
MAS         LDTM,13      MASQUE
            BRU          *32
NMAS        LDTM,13      MASQUE+1
            BRU          *32
      END
```

APENDIX 2

```
C ********************************************************************
C *******                                                        *****
C *******                                                        *****
C *******     I D E N T I F I C A T I O N - S T R I N G          *****
C *******                                                        *****
C *******                                                        *****
C ********************************************************************
C   IDENTIFICATION-STRING IS MEMORY-8 PROGRAM
C          COMPUTATION OF THE REACTIVITY RATIOS IN BINARY
C          IRREVERSIBLE COPOLYMERISATION WITH ULTIMATE EFFECT.
C      INPUT DATA:
C                     THE CARD CONTAINS:
C              NA - NUMBER OF MER A.
C              NB= - NUMBER OF MER B.
C              NJ - NUMBER OF GROUPS A-B.
      DIMENSION T(10000),R(10000)
      DATA A,B/1HA,1HB/
      CALL ASSIGN (4,'CR:')
      READ(4,100) NA,NB,NJ
      CALL ASSIGN (6,'LP')
      N=NA+NB
      CALL GEN(T,R,N)
      DO 3 KA=1,N,5
      DO 3 KB=1,N,5
      CALL GENM(T,R,N,KA,KB,NAB,NAI,NBI)
      IF((NA.GT.NAI-3.AND.NA.LT.NAI+3).AND.
     1(NAB.GT.NJ-3.AND.NAB.LT.NJ+3)) GO TO 5
      GO TO 3
    5 CALL PRINT(T,KA,KB,NAB,NAI,NBI)
    3 CONTINUE
      STOP
```

```
100 FORMAT(3I5)
    END
    SUBROUTINE PRINT(T,KA,KB,KAB,NAI,NBI)
    DIMENSION T(1)
    A=T(KA)
    BVT(KB)
    R1=A/(1.-A)
    R2=(1.-B)/B
    WRITE(6,100) A,B,R1,R2,KAB,NAI,NBI
100 FORMAT(5X,'A=',F10.8,5X,'B=',F10.8,5X,'R1=',F10.5,5X,'R2=',
   1F10.5/
   24X,'NJ=',I5,2X,'NAI=',I5,2X,'NBI=',I5)
    RETURN
    END
    SUBROUTINE GEN(T,R,N)
    DIMENSION T(1),R(1)
    DATA IA/65539/
    DO 1 I=1,100000
    CALL ALEAT(IA,IB,X)
1   IA=IB
    DO 2 I=1,N
    CALL ALEAT(IA,IB,X)
    IA=IB
    T(I)=X
 2  R(I)=X
    CALL RANCRU(T,N,N)
    RETURN
    END
    SUBROUTINE GENN(T,R,N,KA,KB,NAB,NAI,NBI)
    DIMENSION T(1),R(1)
```

```
      NAI=0
      NBI=0
      NAB=0
      IAB=1
      A=T(KA)
      B=T(KB)
      DO 5 I=1,N
      GO TO (1,3),IAB
1     IF(R(I).LE.A) GO TO 2
      NBI=NBI+1
      NAB=NAB+1
      IAB=2
      GO TO 5
2     NAI=NAI+1
      GO TO 5
3     IF(R(I).LE.B) GO TO 4
      NBI=NBI+1
      GO TO 5
4     NAI=NAI+1
      IAB=1
5     CONTINUE
      RETURN
      END
      SUBROUTINE RANCRU(Z,N,NP)
      DIMENSION Z(NP)
      N1=N-1
      IF(Z(1).LE.Z(2)) GO TO 1
      A=Z(2)
      Z(2)=Z(1)
      Z(1)=A
```

```
1   DO 4 I=2,N1
    IF(Z(I).LE.Z(I+1)) GO TO 4
    A=Z(I+1)
    Z(I+1)=Z(I)
    I1=I+1
    DO 2 J=1,I1
    IF(Z(I-J).LE.A) GO TO 3
    Z(I-J+1)=Z(I-J)
2   CONTINUE
    Z(1)=A
    GO TO 4
3   Z(I-J+1)=A
4   CONTINUE
    RETURN
    END
    SUBROUTINE ALEAT(IA,IB,Z)
    DATA I1,I2/0,0/
    Z=RAN(I1,I2)
    RETURN
    END
```