HANDBOOK OF FINITE SEMIGROUP PROGRAMS

John A Hildebrant

TABLE OF CONTENTS

FORWARD

This handbook presents a discussion and gives listings of programs to generate finite semigroups, sort these according to their properties, discover properties of these semigroups, and study their structure from the point of view of the $\mathcal{D}$-class structure, and to determine new structures from these such as the translational hull of a finite semigroup. There are some rather general categories that most of the existing programs known to this author fit into, e.g., properties sorting programs, structure programs, substructure selection programs, and the construction of new semigroups from old.

The programs mentioned in this handbook along with in excess of a hundred others have been developed over time beginning in 1964, and accelerated substantially in 1970 when the algorithm in [Plemmons, 1969] was discovered.

SEMIGROUP GENERATING PROGRAM

We begin with a program to generate the finite semigroups of a given order. There is no input data for this particular program and the output is the collection of all finite semigroups of the specified order in a standard format used in all programs. The semigroup number followed by the order is the first output on a format (1X,I6,I3) followed by the multiplication table for the semigroup in which each row has the format (1X,<N>I2). The algorithm is that suggested in [Plemmons, 1969]. Semigroups listed are the minimal semigroups in each isomorphism (and anti-isomorphism) class. The order given is a lexicographic order for the matrix which constitutes the table. Basically these semigroups are constructed starting with the lowest order; as a new one is considered it is compared with those that have already been generated. If it is isomorphic or anti-isomorphic to one that already exists, then it is discarded; otherwise it is retained in the list.

This program is designed to generate all of the semigroups of a specified order N up to order 8. Considering the fact that there are in excess of 800,000 order 7 semigroups, it is not recommended that this program be used to attempt to generate the semigroups of order 8. It is, however, recommended that this program be changed to generate only those with a certain specified property that will reduce the number of output semigroups substantially.

The output format of this program (gen.f) is such that the list generated by this program can be used as input in any of the other semigroup programs which sort or determine the properties of the semigroups.

The program itself is written in FORTRAN 77 (or any version that will accommodate variable formatting). If a version of fortran is to be used which does not have variable formatting, then the required change in the program simply entails changing format 4 in the main routine to read: 4 FORMAT(1X,5I2) if, for example, the semigroups to be generated are of order 5.

In the open file specification **sem.5** should be changed to sem.x, where x= numerical value of order of the semigroups to be generated.

In the early part of the main program, **N=5** should be changed to N=x, where x=order of the semigroups to be generated.

The table below gives the number of semigroups of orders 3 through 7.

Order	Number of Semigroups
3	18
4	126
5	1160
6	15,973
7	836,021

In view of this table it is not difficult to see that attempting to generate the semigroups of order 8 would not be feasible even with a fast computer and given a long period of time. If we want to appreciate the algorithm established by Plemmons, consider computing the semigroups of order 6 by looking at all of the possible tables and checking to see which are associative, there are 6^{36} or in excess of 10^{28} such tables. This would be a massive undertaking. Consider the problem of generating the semigroups of order 7 in this fashion. There are 7^{49} or in excess of 2.56×10^{41} possible tables. It would not be an unreasonable conjecture that the computer that could do this in a period of a year does not currently exist.

gen.f

```
C Semigroup Generating Program
      INTEGER*2 A(8,8),P(40320,8),Q(40320,8)
      OPEN(6,FILE='sem.5')
    3 FORMAT(1X,I6,I3)
    4 FORMAT(1X,<N>I2)
      N=5
      ID=0
      DO 10 K=1,N
      DO 10 L=1,N
      A(K,L)=1
   10 CONTINUE
      A(N,N)=0
      CALL PERM(P,Q,N,NF)
```

```
      I=N
      J=N
20 CONTINUE
      A(I,J)=A(I,J)+1
      IF(A(I,J).LE.N) GO TO 50
      A(I,J)=0
      IF(J.EQ.1) GO TO 30
      J=J-1
      GO TO 40
30 CONTINUE
      J=N
      I=I-1
40 CONTINUE
      IF(I.EQ.1.AND.J.EQ.1) STOP
      GO TO 20
50 CONTINUE
      CALL ASTEST(A,N,KA)
      IF(KA.EQ.0) GO TO 20
      CALL ISOTEST(A,P,Q,N,NF,KI)
      IF(KI.EQ.0) GO TO 20
      IF(I.EQ.N.AND.J.EQ.N) GO TO 70
      IF(J.EQ.N) GO TO 60
      J=J+1
      GO TO 20
60 J=1
      I=I+1
      GO TO 20
70 CONTINUE
      ID=ID+1
      WRITE(6,3) ID,N
      DO 80 K=1,N
      WRITE(6,4) (A(K,L),L=1,N)
80 CONTINUE
```

```
      GO TO 20
      END

      SUBROUTINE ISOTEST(A,P,Q,N,NF,KI)
      INTEGER*2 A(8,8),B(8,8),C(8,8)
      INTEGER*2 P(40320,8),Q(40320,8)
      KI=0
      L=0
   10 L=L+1
      IF(L.GT.NF) GO TO 50
      DO 30 I=1,N
      DO 30 J=1,N
      NA=A(Q(L,I),Q(L,J))
      IF(NA.EQ.0) GO TO 20
      B(I,J)=P(L,NA)
      GO TO 30
   20 B(I,J)=0
   30 CONTINUE
      CALL COMPARE(A,B,N,KC)
      IF(KC.EQ.0) RETURN
      DO 40 I=1,N
      DO 40 J=1,N
      C(I,J)=B(J,I)
   40 CONTINUE
      CALL COMPARE(A,C,N,KC)
      IF(KC.EQ.0) RETURN
      GO TO 10
   50 CONTINUE
      KI=1
      RETURN
      END

      SUBROUTINE ASTEST(A,N,KA)
      INTEGER*2 A(8,8)
```

```
      KA=1
      I=0
10    I=I+1
      IF(I.GT.N) RETURN
      J=0
20    J=J+1
      IF(J.GT.N) GO TO 10
      MF=A(I,J)
      IF(MF.EQ.0) RETURN
      K=0
30    K=K+1
      IF(K.GT.N) GO TO 20
      MS=A(J,K)
      IF(MS.EQ.0) GO TO 10
      IF(A(I,MS).EQ.0) GO TO 30
      IF(A(MF,K).EQ.0) GO TO 30
      IF(A(I,MS).NE.A(MF,K)) GO TO 40
      GO TO 30
40    KA=0
      RETURN
      END

      SUBROUTINE COMPARE(A,B,N,KC)
      INTEGER*2 A(8,8),B(8,8)
      KC=0
      I=0
10    I=I+1
      IF(I.GT.N) GO TO 30
      J=0
20    J=J+1
      IF(J.GT.N) GO TO 10
      IF(B(I,J).EQ.0) GO TO 30
      IF(A(I,J).GT.B(I,J)) RETURN
```

```
      IF(A(I,J).EQ.B(I,J)) GO TO 20
   30 CONTINUE
      KC=1
      RETURN
      END

      SUBROUTINE PERM(P,Q,N,NF)
      INTEGER*2 P(40320,8),Q(40320,8),T(8)
      J=0
      M=N
      NF=N
   10 M=M-1
      IF(M.EQ.0) GO TO 20
      NF=M*NF
      GO TO 10
   20 CONTINUE
      DO 30 L=1,N
      T(L)=0
   30 CONTINUE
      K=1
      40 CONTINUE
      T(K)=T(K)+1
      IF(T(K).LE.N) GO TO 50
      T(K)=0
      K=K-1
      IF(K.EQ.0) RETURN
      GO TO 40
   50 CONTINUE
      I=0
      60 I=I+1
      IF(I.EQ.K) GO TO 70
      IF(T(I).EQ.T(K)) GO TO 40
      GO TO 60
```

```
 70 CONTINUE
    IF(K.EQ.N) GO TO 80
    K=K+1
    GO TO 40
 80 CONTINUE
    J=J+1
    DO 90 L=1,N
    P(J,L)=T(L)
 90 CONTINUE
    DO 100 L=1,N
    M=P(J,L)
    Q(J,M)=L
100 CONTINUE
    IF(J.EQ.NF) RETURN
    GO TO 40
    END
```

Example. Here is a sample output from the program gen.f with N=5:

```
     393      5
1    1    1    1    1
1    1    1    1    1
3    3    3    3    3
3    3    3    3    3
5    5    5    5    5
```

The number 393 is the identification number of the semigroup and indicates that it was the 393rd semigroup to be generated. Of course, the 5 indicates the order of the semigroup.

COMMUTATIVE SEMIGROUP
SORTING PROGRAM

This program is designed to sort out the commutative semigroups from a list of finite semigroups which are designated as input in standard format. The input format is customized to accept the output of the semigroup generating program above. Name the input file sem.5 if, for example, you are inputing the list of semigroups of order 5, or modify the program input file name to sem.x, where x is the order of the input semigroups.

Input file is sem.5 in the format: The semigroup ID number and order on the first line (1X,I6,I2), followed by the multiplication table in matrix form where the format for each line is (1X,<N>I2) or in this specific case (1X,5I2). We will hereafter refer to this format as **standard format**.

There are two output files. The first com.5 (or com.x where x is the order of the semigroups in the list) will list the commutative semigroups from the input file. The second output file is notcom.5, and will list the semigroups from the input file which are not commutative. Again, both of the output files are in standard format which duplicates that of the input file.

com.f

```
C Commutative Semigroup Sorting Program
      INTEGER*2 S(8,8)
    3 FORMAT(1X,I6,I3)
    4 FORMAT(1X,<N>I2)
      OPEN(5,FILE='sem.5')
      OPEN(6,FILE='com.5')
      OPEN(7,FILE='notcom.5')
   10 CONTINUE
      READ(5,3,END=60) ID,N
      DO 20 I=1,N
      READ(5,4) (S(I,J),J=1,N)
   20 CONTINUE
      CALL ABELS(S,N,KP)
      IF(KP.EQ.0) GO TO 40
```

```
      WRITE(6,3) ID,N
      DO 30 I=1,N
      WRITE(6,4) (S(I,J),J=1,N)
30 CONTINUE
      GO TO 10
40 CONTINUE
      WRITE(7,3) ID,N
      DO 50 I=1,N
      WRITE(7,4) (S(I,J),J=1,N)
50 CONTINUE
      GO TO 10
60 CONTINUE
      STOP
      END

      SUBROUTINE ABELS(S,N,KP)
      INTEGER*2 S(8,8)
      KP=1
      I=0
10 I=I+1
      IF(I.GT.N) RETURN
      J=0
20 J=J+1
      IF(J.GT.N) GO TO 10
      IF(S(I,J).NE.S(J,I)) GO TO 30
      GO TO 20
30 KP=0
      RETURN
      END
```

Example. We display here a sample of the output of com.f. This resulted from the input of a list of order 5 semigroups in sem.5 with the output file com.5. There is, of course, a dual output file notcom.5 which shows those semigroups from the list sem.5 which are not commutative.

683 5

1	1	1	1	1
1	2	1	1	1
1	1	3	1	1
1	1	1	4	4
1	1	1	4	5

Again the number 683 is the identification of the order 5 semigroup which the program selected. It does not mean that there are 682 commutative semigroups on the output list above this one.

This program accepts a list of finite semigroups in standard format and output a detailed description of the Green's class structure. The program gives a breakdown of the $\mathcal{R}$, $\mathcal{L}$, $\mathcal{D}$ and $\mathcal{H}$ classes for each semigroup and an eggbox diagram for each $\mathcal{D}$-class. It also picks out the idempotents.

The program is very useful in attempting to relate properties of the semigroups to their $\mathcal{D}$-class
structure.

green.f

```
C The Green's class structure of a semigroup
      INTEGER*2 S(25,25),D(25,25),H(25,25)
      INTEGER*2 L(25,25),R(25,25)
      INTEGER*2 ND(25),NH(25),NL(25),NR(25)
      OPEN(5,FILE='sem.5')
      OPEN(6,FILE='green.rel')
    3 FORMAT(1X,'D classes')
    4 FORMAT(1X,'H classes')
    5 FORMAT(1X,'L classes')
    6 FORMAT(1X,'R classes')
    7 FORMAT(1X,<M>I3)
    8 FORMAT(1X,' ')
   10 CONTINUE
      CALL LOAD(S,N,LAST)
      IF(LAST.EQ.1) STOP
      CALL IDEM(S,N)
      CALL DCLASS(S,N,D,ND,LD)
      CALL HCLASS(S,N,H,NH,LH)
      CALL LCLASS(S,N,L,NL,LL)
      CALL RCLASS(S,N,R,NR,LR)
      WRITE(6,8)
      WRITE(6,3)
```

```
      DO 20 I=1,LD
      M=ND(I)
      WRITE(6,7) (D(I,J),J=1,M)
20 CONTINUE
      WRITE(6,8)
      WRITE(6,4)
      DO 30 I=1,LH
      M=NH(I)
      WRITE(6,7) (H(I,J),J=1,M)
30 CONTINUE
      WRITE(6,8)
      WRITE(6,5)
      DO 40 I=1,LL
      M=NL(I)
      WRITE(6,7) (L(I,J),J=1,M)
40 CONTINUE
      WRITE(6,8)
      WRITE(6,6)
      DO 50 I=1,LR
      M=NR(I)
      WRITE(6,7) (R(I,J),J=1,M)
50 CONTINUE
      CALL BOX(S,N,D,ND,LD,H,NH,LH)
      GO TO 10
      END

      SUBROUTINE DCLASS(S,N,D,ND,LD)
      INTEGER*2 S(25,25),D(25,25),ND(25)
      INTEGER*2 Q(25),DM(25,25)
      CALL DMAT(S,N,DM)
      LD=0
      DO 10 I=1,N
      Q(I)=0
```

```
10 CONTINUE
   DO 30 I=1,N
   IF(Q(I).EQ.1) GO TO 30
   LD=LD+1
   Q(I)=1
   NT=0
   DO 20 J=I,N
   IF(DM(I,J).EQ.0) GO TO 20
   NT=NT+1
   D(LD,NT)=J
   Q(J)=1
20 CONTINUE
   ND(LD)=NT
30 CONTINUE
   RETURN
   END

   SUBROUTINE HCLASS(S,N,H,NH,LH)
   INTEGER*2 S(25,25),H(25,25),NH(25)
   INTEGER*2 Q(25),HM(25,25)
   CALL HMAT(S,N,HM)
   LH=0
   DO 10 I=1,N
   Q(I)=0
10 CONTINUE
   DO 30 I=1,N
   IF(Q(I).EQ.1) GO TO 30
   LH=LH+1
   Q(I)=1
   NT=0
   DO 20 J=I,N
   IF(HM(I,J).EQ.0) GO TO 20
   NT=NT+1
```

```
      H(LH,NT)=J
      Q(J)=1
   20 CONTINUE
      NH(LH)=NT
   30 CONTINUE
      DO 50 I=1,LH
      M=NH(I)
      IF(M.EQ.1) GO TO 50
      L=0
   40 L=L+1
      IF(L.GT.M) GO TO 50
      KX=H(I,L)
      IF(S(KX,KX).NE.KX) GO TO 40
      H(I,L)=H(I,1)
      H(I,1)=KX
   50 CONTINUE
      RETURN
      END

      SUBROUTINE LCLASS(S,N,L,NL,LL)
      INTEGER*2 S(25,25),L(25,25),NL(25)
      INTEGER*2 Q(25),LM(25,25)
      CALL LMAT(S,N,LM)
      LL=0
      DO 10 I=1,N
      Q(I)=0
   10 CONTINUE
      DO 30 I=1,N
      IF(Q(I).EQ.1) GO TO 30
      LL=LL+1
      Q(I)=1
      NT=0
      DO 20 J=I,N
```

```
      IF(LM(I,J).EQ.0) GO TO 20
      NT=NT+1
      L(LL,NT)=J
      Q(J)=1
   20 CONTINUE
      NL(LL)=NT
   30 CONTINUE
      RETURN
      END

      SUBROUTINE RCLASS(S,N,R,NR,LR)
      INTEGER*2 S(25,25),R(25,25),NR(25)
      INTEGER*2 Q(25),RM(25,25)
      CALL RMAT(S,N,RM)
      LR=0
      DO 10 I=1,N
      Q(I)=0
   10 CONTINUE
      DO 30 I=1,N
      IF(Q(I).EQ.1) GO TO 30
      LR=LR+1
      Q(I)=1
      NT=0
      DO 20 J=I,N
      IF(RM(I,J).EQ.0) GO TO 20
      NT=NT+1
      R(LR,NT)=J
      Q(J)=1
   20 CONTINUE
      NR(LR)=NT
   30 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE LOAD(S,N,LAST)
      INTEGER*2 S(25,25)
    3 FORMAT(1X,I6,I3)
    4 FORMAT(1X,<N>I2)
    5 FORMAT(1X,<N>I3)
    6 FORMAT('*')
      LAST=0
      READ(5,3,END=20) IDS,N
      WRITE(6,6)
      WRITE(6,3) IDS,N
      DO 10 I=1,N
      IF(N.LE.9) READ(5,4) (S(I,J),J=1,N)
      IF(N.LE.9) WRITE(6,4) (S(I,J),J=1,N)
      IF(N.GE.10) READ(5,5) (S(I,J),J=1,N)
      IF(N.GE.10) WRITE(6,5) (S(I,J),J=1,N)
   10 CONTINUE
      RETURN
   20 LAST=1
      RETURN
      END

      SUBROUTINE DREL(S,N,I,J,KF)
      INTEGER*2 S(25,25)
      K=0
      KF=0
   10 K=K+1
      IF(K.GT.N) RETURN
      CALL LREL(S,N,I,K,KL)
      IF(KL.EQ.0) GO TO 10
      CALL RREL(S,N,J,K,KR)
      IF(KR.EQ.0) GO TO 10
      KF=1
      RETURN
```

```
      END

      SUBROUTINE HREL(S,N,I,J,KF)
      INTEGER*2 S(25,25)
      KF=0
      CALL LREL(S,N,I,J,KL)
      IF(KL.EQ.0) RETURN
      CALL RREL(S,N,I,J,KR)
      IF(KR.EQ.0) RETURN
      KF=1
      RETURN
      END

      SUBROUTINE LREL(S,N,I,J,KF)
      INTEGER*2 S(25,25)
      KF=1
      IF(I.EQ.J) RETURN
      KF=0
      K=0
   10 K=K+1
      IF(K.GT.N) GO TO 20
      IF(S(K,J).NE.I) GO TO 10
      KF=1
   20 CONTINUE
      IF(KF.EQ.0) RETURN
      KF=0
      K=0
   30 K=K+1
      IF(K.GT.N) RETURN
      IF(S(K,I).NE.J) GO TO 30
      KF=1
      RETURN
      END
```

```
      SUBROUTINE RREL(S,N,I,J,KF)
      INTEGER*2 S(25,25)
      KF=1
      IF(I.EQ.J) RETURN
      KF=0
      K=0
   10 K=K+1
      IF(K.GT.N) GO TO 20
      IF(S(J,K).NE.I) GO TO 10
      KF=1
   20 CONTINUE
      IF(KF.EQ.0) RETURN
      KF=0
      K=0
   30 K=K+1
      IF(K.GT.N) RETURN
      IF(S(I,K).NE.J) GO TO 30
      KF=1
      RETURN
      END

      SUBROUTINE DMAT(S,N,DM)
      INTEGER*2 S(25,25),DM(25,25)
      DO 10 I=1,N
      DO 10 J=1,N
      CALL DREL(S,N,I,J,KF)
      DM(I,J)=KF
   10 CONTINUE
      RETURN
      END

      SUBROUTINE HMAT(S,N,HM)
      INTEGER*2 S(25,25),HM(25,25)
      DO 10 I=1,N
```

```
      DO 10 J=1,N
      CALL HREL(S,N,I,J,KF)
      HM(I,J)=KF
   10 CONTINUE
      RETURN
      END
      SUBROUTINE LMAT(S,N,LM)
      INTEGER*2 S(25,25),LM(25,25)
      DO 10 I=1,N
      DO 10 J=1,N
      CALL LREL(S,N,I,J,KF)
      LM(I,J)=KF
   10 CONTINUE
      RETURN
      END

      SUBROUTINE RMAT(S,N,RM)
      INTEGER*2 S(25,25),RM(25,25)
      DO 10 I=1,N
      DO 10 J=1,N
      CALL RREL(S,N,I,J,KF)
      RM(I,J)=KF
   10 CONTINUE
      RETURN
      END

      SUBROUTINE IDEM(S,N)
      INTEGER*2 S(25,25),ID(25)
    4 FORMAT(1X,' ')
    5 FORMAT(1X,'Idempotents',<M>I3)
      WRITE(6,4)
      NID=0
      DO 10 I=1,N
      IF(S(I,I).NE.I) GO TO 10
```

```
      NID=NID+1
      ID(NID)=I
   10 CONTINUE
      M=NID
      WRITE(6,5) (ID(J),J=1,M)
      RETURN
      END

      SUBROUTINE BOX(S,N,D,ND,LD,H,NH,LH)
      INTEGER*2 S(25,25),A(25,25),B(25)
      INTEGER*2 D(25,25),ND(25),H(25,25),NH(25)
    3 FORMAT(1X,' ')
    4 FORMAT(1X,'Egg Box Diagram')
    5 FORMAT(1X,<NTAB>X,<NOC>I3)
      NTAB=1
      WRITE(6,3)
      WRITE(6,4)
      WRITE(6,3)
      DO 10 I=1,N
      B(I)=1
   10 CONTINUE
      DO 50 I=1,LH
      M=NH(I)
      IF(M.EQ.1) GO TO 50
      DO 40 J=2,M
      B(H(I,J))=0
   40 CONTINUE
   50 CONTINUE
      DO 150 KD=1,LD
      DO 60 I=1,N
      DO 60 J=1,N
      A(I,J)=0
   60 CONTINUE
```

```
      I=0
      NOD=ND(KD)
   70 I=I+1
      IF(I.GT.LH) GO TO 90
      IF(B(H(I,1)).EQ.0) GO TO 70
      L=0
   80 L=L+1
      IF(L.GT.NOD) GO TO 70
      IF(H(I,1).NE.D(KD,L)) GO TO 80
      A(1,1)=H(I,1)
      B(A(1,1))=0
   90 CONTINUE
      NOC=1
      MA=A(1,1)
      DO 100 J=1,N
      IF(B(J).EQ.0) GO TO 100
      CALL RREL(S,N,J,MA,KF)
      IF(KF.EQ.0) GO TO 100
      NOC=NOC+1
      A(1,NOC)=J
      B(J)=0
  100 CONTINUE
      NOR=1
      DO 110 I=1,N
      IF(B(I).EQ.0) GO TO 110
      CALL LREL(S,N,I,MA,KF)
      IF(KF.EQ.0) GO TO 110
      NOR=NOR+1
      A(NOR,1)=I
      B(I)=0
  110 CONTINUE
      IF(NOC.EQ.1.OR.NOR.EQ.1) GO TO 135
      DO 130 I=2,NOR
```

```
      DO 130 J=2,NOC
      DO 120 K=1,N
      IF(B(K).EQ.0) GO TO 120
      NA=A(I,1)
      CALL RREL(S,N,NA,K,KF)
      IF(KF.EQ.0) GO TO 120
      NB=A(1,J)
      CALL LREL(S,N,NB,K,KF)
      IF(KF.EQ.0) GO TO 120
      A(I,J)=K
      B(K)=0
  120 CONTINUE
  130 CONTINUE
  135 CONTINUE
      WRITE(6,3)
      DO 140 I=1,NOR
      WRITE(6,5) (A(I,J),J=1,NOC)
  140 CONTINUE
      NTAB=NTAB+3*NOC+1
  150 CONTINUE
      RETURN
      END
```

Example. Here is a sample of the output of green.f with the input being semigroup number 884 of order 5. The output consists of a display of the input semigroup, its idempotents, and its $\mathcal{D}$-classes, $\mathcal{H}$-classes, $\mathcal{L}$-classes, and $\mathcal{R}$-classes, followed by a display of the egg box diagram.

884 5

1	1	1	1	1
1	2	2	2	2
1	2	2	2	3
1	4	4	4	4
1	2	2	4	5

Idempotents 1 2 4 5

D classes
1
2 4
3
5

H classes
1
2
3
4
5

L classes
1
2 4
3
5

R classes
1
2
3
4
5

Egg Box Diagram

1

2
4
3
5

The tran.f program is designed to determine the translational hull of a finite semigroup. There are limits imposed by the program on the size of the translational hull that will be output. It is currently set at 25. If this limit is exceeded, the program will still display some information e.g., the number of bitranslations. Each input semigroup is tested to determine whether the semigroup is weakly reductive (in which case it is a subsemigroup of its translational hull) and whether it is abelian (commutative). The translational degree is revealed if the semigroup is weakly reductive, i.e., the number of bitranslations added onto the semigroup to obtain the translational hull. The linked pairs are displayed, along with the input semigroup and the translational hull. All of the above information is output in the file **tran.ans**. The input file is e.g., **sem.5** in standard format. There is a second output file **tran.dt** which simply outputs the translational hull in standard format in case it is required for a later program.

tran.f

```
C Computes the translational hull
      INTEGER*2 S(15,15)
      OPEN(10,FILE='sem.5')
      OPEN(6,FILE='tran.ans')
      OPEN(7,FILE='tran.dt')
      LIM=25
      KSTP=0
   10 CONTINUE
      CALL LOAD(S,LIM,KSTP)
      IF(KSTP.EQ.1) STOP
      GO TO 10
      END

      SUBROUTINE LOAD(S,LIM,KSTP)
      INTEGER*2 S(15,15),ML(3000,15)
      INTEGER*2 MR(3000,15)
    3 FORMAT(1X,I6,I3)
```

```
 4 FORMAT(1X,<N>I2)
 5 FORMAT(1X,¡N¿I3)
 6 FORMAT('*')
   WRITE(6,6)
   READ(10,3,END=50) K,N
   KC=0
   DO 10 I=1,N
   IF(N.LE.9) READ(10,4) (S(I,J),J=1,N)
   IF(N.GE.10) READ(10,5) (S(I,J),J=1,N)
10 CONTINUE
   IF(KC.NE.0) RETURN
   WRITE(6,3) K,N
   DO 20 I=1,N
   IF(N.LE.9) WRITE(6,4) (S(I,J),J=1,N)
   IF(N.GE.10) WRITE(6,5) (S(I,J),J=1,N)
20 CONTINUE
   CALL NEXT(S,N,ML,MR,NL,NR,LIM)
   IF(NL.GT.LIM) RETURN
   IF(NR.GT.LIM) RETURN
   IV=K
   CALL LAST(S,N,ML,MR,NL,NR,IV,LIM)
   RETURN
50 KSTP=1
   RETURN
   END

   SUBROUTINE LAST(S,N,ML,MR,NL,NR,IV,
   LIM)
   INTEGER*2 S(15,15),ML(3000,15)
   INTEGER*2 MR(3000,15)
   INTEGER*2 L(15),R(15)
   INTEGER*2 HL(3000,15),HR(3000,15)
 3 FORMAT(1X,'Translation Number',I3,2X,'Left')
```

```
 4 FORMAT(1X,'Translation Number',I3,2X,'Right')
 5 FORMAT(1X,' ')
 6 FORMAT(1X,'Translations')
 7 FORMAT(1X,6X,'Left',<N-2>X,<N-2>X,3X,'Right')
 8 FORMAT(<N>I2)
 9 FORMAT(1X,'Num',1X,'Left',I4,2X,'Right', I4,2X, 'Trans',I4)
   NSUM=0
   DO 30 I=1,NL
   DO 10 K=1,N
10 L(K)=ML(I,K)
   DO 30 J=1,NR
   DO 20 K=1,N
20 R(K)=MR(J,K)
   CALL LINK(S,N,L,R,KF)
   IF(KF.EQ.0) GO TO 30
   NSUM=NSUM+1
   IF(NSUM.GT.LIM) GO TO 30
   DO 25 NH=1,N
   HL(NSUM,NH)=L(NH)
   HR(NSUM,NH)=R(NH)
25 CONTINUE
30 CONTINUE
   WRITE(6,9) NL,NR,NSUM
   NG=N
   CALL WEAK(S,NG,KFG,NSUM)
   CALL IDENT(S,NG,KIG)
   CALL ABEL(S,N,KA)
   IF(NSUM.GT.LIM) GO TO 50
   CALL SWID(HL,N,HR,NSUM)
   CALL SWAP(S,N,HL,HR,NSUM)
   WRITE(6,6)
   WRITE(6,7)
   DO 35 NP=1,NSUM
```

```
      WRITE(6,3) NP
      WRITE(6,8) (HL(NP,K1),K1=1,N)
      WRITE(6,4) NP
      WRITE(6,8) (HR(NP,K2),K2=1,N)
   35 CONTINUE
      CALL BUILD(S,N,HL,HR,NSUM,IV)
   50 CONTINUE
      WRITE(6,5)
      WRITE(6,5)
      WRITE(6,5)
      RETURN
      END

      SUBROUTINE LINK(S,N,L,R,KF)
      INTEGER*2 S(15,15),L(15),R(15)
      KF=1
      I=0
      NP=N+1
   10 I=I+1
      IF(I.EQ.NP) RETURN
      J=0
   20 J=J+1
      IF(J.EQ.NP) GO TO 10
      NA=L(J)
      NB=R(I)
      NB=S(I,NA)-S(NB,J)
      IF(NB.EQ.0) GO TO 20
      KF=0
      RETURN
      END

      SUBROUTINE PRINT(M,NSUM,IV)
      INTEGER*2 M(50,50)
    2 FORMAT(1X,'Translational Hull')
```

```
 4 FORMAT(1X,I6,I3)
 5 FORMAT(1X,<NSUM>I2)
 6 FORMAT(1X,<NSUM>I3)
20 FORMAT(1X,20I3)
   WRITE(6,2)
   IF(NSUM.GT.25) RETURN
   CONTINUE
   WRITE(7,4) IV,NSUM
   DO 50 I=1,NSUM
   WRITE(6,20) (M(I,J),J=1,NSUM)
   IF(NSUM.LE.9) WRITE(7,5)
   (M(I,J),J=1,NSUM)
   IF(NSUM.GE.10) WRITE(7,6)
   (M(I,J),J=1,NSUM)
50 CONTINUE
   RETURN
   END

   SUBROUTINE BUILD(S,N,HL,HR,NSUM,IV)
   INTEGER*2 S(15,15),HL(3000,15),HR(3000,15)
   INTEGER*2 M(50,50)
   I=0
10 I=I+1
   IF(I.GT.NSUM) GO TO 60
   J=0
20 J=J+1
   IF(J.GT.NSUM) GO TO 10
   K=0
30 K=K+1
   IF(K.GT.NSUM) GO TO 20
   MA=0
   DO 50 NP=1,N
   MB=HL(J,NP)
```

```
      MB=HL(I,MB)-HL(K,NP)
      MC=HR(I,NP)
      MC=HR(J,MC)-HR(K,NP)
      MA=MA+MB*MB+MC*MC
   50 CONTINUE
      IF(MA.EQ.0) M(I,J)=K
      GO TO 30
   60 CONTINUE
      CALL PRINT(M,NSUM,IV)
      RETURN
      END

      SUBROUTINE SWAP(S,N,HL,HR,NSUM)
      INTEGER*2 S(15,15),HL(3000,15),HR(3000,15)
      DO 50 I=1,N
      K=0
   10 K=K+1
      IF(K.GT.NSUM) GO TO 50
      MA=0
      DO 20 J=1,N
      NA=HL(K,J)-S(I,J)
      NB=HR(K,J)-S(J,I)
      MA=MA+NA*NA+NB*NB
   20 CONTINUE
      IF(MA.NE.0) GO TO 10
      CONTINUE
      DO 30 J=1,N
      NP=HL(I,J)
      HL(I,J)=HL(K,J)
      HL(K,J)=NP
      NP=HR(I,J)
      HR(I,J)=HR(K,J)
      HR(K,J)=NP
```

```
30 CONTINUE
50 CONTINUE
   RETURN
   END

   SUBROUTINE SWID(HL,N,HR,NSUM)
   INTEGER*2 HL(3000,15),HR(3000,15)
   I=0
10 I=I+1
   IF(I.GT.NSUM) RETURN
   NP=0
   DO 20 J=1,N
   NA=HL(I,J)-J
   NB=HR(I,J)-J
   NP=NP+NA*NA+NB*NB
20 CONTINUE
   IF(NP.NE.0) GO TO 10
   DO 30 J=1,N
   NT=HL(I,J)
   HL(I,J)=HL(NSUM,J)
   HL(NSUM,J)=NT
   NT=HR(I,J)
   HR(I,J)=HR(NSUM,J)
   HR(NSUM,J)=NT
30 CONTINUE
   RETURN
   END

   SUBROUTINE WEAK(S,N,KF,NSUM)
   INTEGER*2 S(15,15)
 9 FORMAT(1X,10X,'Weakly Reductive',7X,
   'Translational Degree',I3)
   IDEG=NSUM-N
   KF=1
```

```
      J=0
   10 J=J+1
      IF(J.EQ.N) GO TO 40
      JP=J+1
      DO 30 I=JP,N
      NC=0
      ND=0
      DO 20 K=1,N
      NA=S(I,K)-S(J,K)
      NB=S(K,I)-S(K,J)
      NC=NC+NA*NA
      ND=ND+NB*NB
   20 CONTINUE
      IF(NC.EQ.0.AND.ND.EQ.0) KF=0
   30 CONTINUE
      IF(KF.NE.0) GO TO 10
      CONTINUE
      RETURN
   40 CONTINUE
      WRITE(6,9) IDEG
      RETURN
      END

      SUBROUTINE IDENT(S,N,KI)
      INTEGER*2 S(15,15)
    9 FORMAT(1X,10X,'Identity',1X,I3)
      KI=0
      I=0
   10 I=I+1
      IF(I.GT.N) GO TO 30
      NS=0
      DO 20 J=1,N
      NA=S(I,J)-J
```

```
      NB=S(J,I)-J
      NS=NS+NA*NA+NB*NB
   20 CONTINUE
      IF(NS.NE.0) GO TO 10
      KI=1
      WRITE(6,9) I
      RETURN
   30 CONTINUE
      RETURN
      END

      SUBROUTINE ABEL(S,N,KA)
    3 FORMAT(1X,10X,'Abelian')
      INTEGER*2 S(15,15)
      KA=1
      I=0
   10 I=I+1
      IF(I.GT.N) GO TO 40
      J=0
   20 J=J+1
      IF(J.GT.N) GO TO 10
      IF(S(I,J).NE.S(J,I)) GO TO 30
      GO TO 20
   30 KA=0
      RETURN
   40 CONTINUE
      WRITE(6,3)
      RETURN
      END

      SUBROUTINE TEST(S,N,T,KF,ID)
      INTEGER*2 S(15,15),T(15)
      KF=0
      I=0
```

```
10 I=I+1
   IF(I.GT.N) GO TO 40
   J=0
20 J=J+1
   IF(J.GT.N) GO TO 10
   IF(ID.EQ.1) GO TO 30
   IF(T(S(I,J)).NE.S(T(I),J)) RETURN
   GO TO 20
30 CONTINUE
   IF(T(S(I,J)).NE.S(I,T(J))) RETURN
   GO TO 20
40 KF=1
   RETURN
   END

   SUBROUTINE NEXT(S,N,ML,MR,NL,NR,
   LIM)
   INTEGER*2 S(15,15),ML(3000,15)
   INTEGER*2 MR(3000,15)
   INTEGER*2 T(15),M(15)
   NL=0
   NR=0
   DO 5 I=1,15
   M(I)=N
   IF(N.LT.I) M(I)=1
 5 CONTINUE
   DO 60 L15=1,M(15)
   T(15)=L15
   DO 60 L14=1,M(14)
   T(14)=L14
   DO 60 L13=1,M(13)
   T(13)=L13
   DO 60 L12=1,M(12)
```

```
      T(12)=L12
      DO 60 L11=1,M(11)
      T(11)=L11
      DO 60 L10=1,M(10)
      T(10)=L10
      DO 60 L9=1,M(9)
      T(9)=L9
      DO 60 L8=1,M(8)
      T(8)=L8
      DO 60 L7=1,M(7)
      T(7)=L7
      DO 60 L6=1,M(6)
      T(6)=L6
      DO 60 L5=1,M(5)
      T(5)=L5
      DO 60 L4=1,M(4)
      T(4)=L4
      DO 60 L3=1,M(3)
      T(3)=L3
      DO 60 L2=1,M(2)
      T(2)=L2
      DO 60 L1=1,M(1)
      T(1)=L1
      ID=0
      CALL TEST(S,N,T,KF,ID)
      IF(KF.EQ.0) GO TO 20
      NL=NL+1
      IF(NL.GT.LIM) RETURN
      DO 10 I=1,N
   10 ML(NL,I)=T(I)
   20 CONTINUE
      ID=1
      CALL TEST(S,N,T,KF,ID)
```

```
      IF(KF.EQ.0) GO TO 50
      NR=NR+1
      IF(NR.GT.LIM) RETURN
      DO 30 I=1,N
   30 MR(NR,I)=T(I)
   50 CONTINUE
   60 CONTINUE
      RETURN
      END
```

Example. We present an example of the information that the program yields for a given input semigroup. To fit the information into the format of this handbook, we will not conform to the output format of tran.f in presenting this example, although the information displayed matches that of the program. The semigroup under consideration is weakly reductive and hence is a subsemigroup of the translational hull. The input semigroup is an order 5 semigroup and the translational hull is order 8. We will display, as the program does, the linked pairs of left and right translations which constitute the 8 bitranslations of the translational hull and number them to correspond to the elements 1 through 8 in the multiplication table.

The input order 5 semigroup is 1113 5 and its table is:

1	1	1	4	5
1	1	1	4	5
1	2	3	4	5
4	4	4	5	1
5	5	5	1	4

Observe that the semigroup is not commutative, so we anticipate that some of the left translations will be linked with a right translation different from itself.

No	Left	Right
1	1 1 1 4 5	1 1 1 4 5
2	1 1 1 4 5	1 1 2 4 5
3	1 2 3 4 5	1 1 3 4 5
4	4 4 4 5 1	4 4 4 5 1
5	5 5 5 1 4	5 5 5 1 4
6	1 1 1 4 5	1 2 2 4 5
7	1 1 1 4 5	1 2 1 4 5
8	1 2 3 4 5	1 2 3 4 5

The translational hull is the semigroup:

1	1	1	4	5	1	1	1
1	1	1	4	5	2	2	2
1	2	3	4	5	2	1	3
4	4	4	5	1	4	4	4
5	5	5	1	4	5	5	5
1	1	1	4	5	6	6	6
1	1	1	4	5	7	7	7
1	2	3	4	5	6	7	8

Observe the input semigroup in the upper left corner of the matrix.

The program hom.f accepts two semigroups S followed by a commutative semigroup T with both in standard format. This input pattern can be continued for additional pairs. The output is the semigroup of homomorphisms $\mathrm{Hom}(S,T)$.

hom.f

```
C Computes Hom(S,T) for semigroups S and T
C The semigroup T is required to be commutative.
      INTEGER*2 S(25,25),T(25,25),H(1500,25),Q(1500,1500)
      OPEN(10,FILE='sem.in')
      OPEN(6,FILE='hom.ans')
      OPEN(7,FILE='hom.dt')
    6 CONTINUE
      CALL LOAD(S,N,IDS,T,M,IDT)
      CALL GEN(S,N,T,M,H,IC,KF)
      IF(KF.EQ.0) GO TO 10
      CALL MULT(T,M,H,IC,Q)
   10 CONTINUE
      CALL DUMP(S,N,IDS,T,M,IDT,H,IC,Q,KF)
      GO TO 6
      END

      SUBROUTINE LOAD(S,N,IDS,T,M,IDT)
      INTEGER*2 S(25,25),T(25,25)
    2 FORMAT(1X,I6,I3)
    3 FORMAT(1X,¡MN¿I2)
    4 FORMAT(1X,¡MN¿I3)
    5 CONTINUE
      READ(10,2,END=30) IDS,N
      MN=N
      DO 10 I=1,N
      IF(N.LE.9) READ(10,3) (S(I,J),J=1,N)
      IF(N.GE.10) READ(10,4) (S(I,J),J=1,N)
```

```
10 CONTINUE
   READ(10,2) IDT,M
   MN=M
   DO 20 I=1,M
   IF(M.LE.9) READ(10,3) (T(I,J),J=1,M)
   IF(M.GE.10) READ(10,4) (T(I,J),J=1,M)
20 CONTINUE
   RETURN
30 CONTINUE
   STOP
   END

   SUBROUTINE GEN(S,N,T,M,H,IC,KF)
   INTEGER*2 S(25,25),T(25,25),P(25),H(1500,25)
   IC=0
   KF=1
   GO TO (9,8,7,6,5,4,3,2,1),N
 1 L9=0
21 L9=L9+1
   IF(L9.GT.M) GO TO 40
   P(9)=L9
 2 L8=0
22 L8=L8+1
   IF(L8.GT.M.AND.N.GT.8) GO TO 21
   IF(L8.GT.M) GO TO 40
   P(8)=L8
 3 L7=0
23 L7=L7+1
   IF(L7.GT.M.AND.N.GT.7) GO TO 22
   IF(L7.GT.M) GO TO 40
   P(7)=L7
 4 L6=0
24 L6=L6+1
```

```
      IF(L6.GT.M.AND.N.GT.6) GO TO 23
      IF(L6.GT.M) GO TO 40
      P(6)=L6
    5 L5=0
   25 L5=L5+1
      IF(L5.GT.M.AND.N.GT.5) GO TO 24
      IF(L5.GT.M) GO TO 40
      P(5)=L5
    6 L4=0
   26 L4=L4+1
      IF(L4.GT.M.AND.N.GT.4) GO TO 25
      IF(L4.GT.M) GO TO 40
      P(4)=L4
    7 L3=0
   27 L3=L3+1
      IF(L3.GT.M.AND.N.GT.3) GO TO 26
      IF(L3.GT.M) GO TO 40
      P(3)=L3
    8 L2=0
   28 L2=L2+1
      IF(L2.GT.M.AND.N.GT.2) GO TO 27
      IF(L2.GT.M) GO TO 40
      P(2)=L2
    9 L1=0
   29 L1=L1+1
      IF(L1.GT.M.AND.N.GT.1) GO TO 28
      IF(L1.GT.M) GO TO 40
      P(1)=L1
      CALL TEST(S,N,T,M,P,KH)
      IF(KH.EQ.0) GO TO 29
      CALL HOM(H,IC,P,N)
      IF(IC.GT.1500) GO TO 50
      GO TO 29
```

```
40 CONTINUE
   RETURN
50 KF=0
   RETURN
   END

   SUBROUTINE TEST(S,N,T,M,P,KH)
   INTEGER*2 S(25,25),T(25,25),P(25)
   KH=1
   I=0
10 I=I+1
   IF(I.GT.N) GO TO 40
   J=0
20 J=J+1
   IF(J.GT.N) GO TO 10
   IF(P(S(I,J)).NE.T(P(I),P(J))) GO TO 30
   GO TO 20
30 KH=0
40 CONTINUE
   RETURN
   END

   SUBROUTINE HOM(H,IC,P,N)
   INTEGER*2 H(1500,25),P(25)
   IC=IC+1
   IF(IC.GT.1500) GO TO 20
   DO 10 I=1,N
10 H(IC,I)=P(I)
   RETURN
20 CONTINUE
   RETURN
   END

   SUBROUTINE MULT(T,M,H,IC,Q)
```

```
      INTEGER*2 T(25,25),H(1500,25),Q(1500,1500)
      DO 40 IA=1,IC
      DO 40 IB=1,IC
      I=0
   10 I=I+1
      J=0
   20 J=J+1
      IF(J.GT.M) GO TO 30
      IF(T(H(IA,J),H(IB,J)).NE.H(I,J)) GO TO 10
      GO TO 20
   30 Q(IA,IB)=I
   40 CONTINUE
      RETURN
      END

      SUBROUTINE DUMP(S,N,IDS,T,M,IDT,H,IC,Q,KF)
      INTEGER*2 S(25,25),T(25,25),H(1500,25),Q(1500,1500)
    2 FORMAT(1X,I3,3X,<N>I3)
    3 FORMAT(1X,I6,I3)
    4 FORMAT(1X,< L >I2)
    5 FORMAT(1X,< L >I3)
    6 FORMAT(1X,' ')
    7 FORMAT(1X,'Semigroup S')
    8 FORMAT(1X,'Semigroup T')
    9 FORMAT(1X,'Homomorphisms')
   10 FORMAT(1X,'HOM(S,T)')
  100 FORMAT('*')
      WRITE(6,6)
      WRITE(6,6)
      WRITE(6,100)
      WRITE(6,6)
      WRITE(6,6)
      WRITE(6,6)
```

```
      WRITE(6,6)
      WRITE(6,7)
      WRITE(6,3) IDS,N
      L=N
      DO 20 I=1,N
      IF(N.LE.9) WRITE(6,4) (S(I,J),J=1,N)
      IF(N.GE.10) WRITE(6,5) (S(I,J),J=1,N)
   20 CONTINUE
      WRITE(6,8)
      WRITE(6,3) IDT,M
      L=M
      DO 30 I=1,M
      IF(M.LE.9) WRITE(6,4) (T(I,J),J=1,M)
      IF (M.GE.10) WRITE(6,5) (T(I,J),J=1,M)
   30 CONTINUE
      IF(KF.EQ.0) RETURN
      WRITE(6,9)
      DO 40 I=1,IC
   40 WRITE(6,2) I,(H(I,J),J=1,N)
      WRITE(6,10)
      WRITE(7,3) IDS,IC
      L=IC
      DO 50 I=1,IC
      IF(IC.LE.9) WRITE(6,4) (Q(I,J),J=1,IC)
      IF(IC.GE.10) WRITE(6,5) (Q(I,J),J=1,IC)
      IF(IC.LE.9) WRITE(7,4) (Q(I,J),J=1,IC)
      IF(IC.GE.10) WRITE(7,5) (Q(I,J),J=1,IC)
   50 CONTINUE
      RETURN
      END
```

Example. We present an example from the output of the hom.f program. The input for the result was the two semigroups S and T. Homomorphism are listed

according to the number in which they appear in the matrix for $\mathrm{Hom}(S,T)$.

Semigroup S

1151 5

1	1	3	4	5
1	2	3	4	5
3	3	1	5	4
4	4	5	1	3
5	5	4	3	1

Semigroup T

122 4

1	1	3	4
1	2	3	4
3	3	4	1
4	4	1	3

Homomorphisms

1	1	1	1	1	1
2	1	2	1	1	1
3	2	2	2	2	2

HOM(S,T)

1	1	1
1	2	2
1	2	3

WEAKLY REDUCTIVE SEMIGROUP SORTING PROGRAM

The program wred.f sorts an input semigroup list in standard format according to whether the semigroup is weakly reductive. The input file is sem.5, and as usual the program will accommodate any order semigroup up to order 8 and the program can be modified to give the input file a name accordingly. The two output files are wred.dt if the semigroup is weakly reductive, and notwred.dt if it is not. The output in both files appears in standard format.

An important feature of weakly reductive semigroups has a strong connection to the program above which determines the translational hull of a semigroup. It is well known that a semigroup is injective in its translational hull if and only if it is weakly reductive.

wred.f

```
C Weakly Reductive Semigroup Sorting Program
      INTEGER*2 S(8,8)
    3 FORMAT(1X,I6,I3)
    4 FORMAT(1X,¡N¿I2)
      OPEN(5,FILE='sem.5')
      OPEN(6,FILE='wred.dt')
      OPEN(7,FILE='notwred.dt')
   10 CONTINUE
      READ(5,3,END=60) ID,N
      DO 20 I=1,N
      READ(5,4) (S(I,J),J=1,N)
   20 CONTINUE
      CALL WKRED(S,N,KP)
      IF(KP.EQ.0) GO TO 40
      WRITE(6,3) ID,N
      DO 30 I=1,N
      WRITE(6,4) (S(I,J),J=1,N)
   30 CONTINUE
```

```
      GO TO 10
40    CONTINUE
      WRITE(7,3) ID,N
      DO 50 I=1,N
      WRITE(7,4) (S(I,J),J=1,N)
50    CONTINUE
      GO TO 10
60    CONTINUE
      STOP
      END

      SUBROUTINE WKRED(S,N,KP)
      INTEGER*2 S(8,8)
      KP=1
      J=0
10    J=J+1
      IF(J.EQ.N) RETURN
      I=J
20    I=I+1
      IF(I.GT.N) GO TO 10
      N1=0
      N2=0
      DO 30 K=1,N
      M1=S(I,K)-S(J,K)
      M2=S(K,I)-S(K,J)
      N1=N1+M1*M1
      N2=N2+M2*M2
30    CONTINUE
      IF(N1.EQ.0.AND.N2.EQ.0) GO TO 40
      GO TO 20
40    KP=0
      RETURN
      END
```

We will not present a sample output here, since the output into both files is in standard format, which means it can be used in other programs.

LATTICE OF CONGRUENCES PROGRAM

The program latcong.f inputs a list of semigroups in standard format and determines the lattice of congruences. Input for this is a list of semigroups in standard format in a file named sem.in. There are three output files. The first is latcong.ans, which lists the input semigroup and the relation matrices for the congruences on the input semigroup, followed by the matrix for the lattice of congruences. The final output file indicates that the congruences form a chain.

latcong.f

```
C Displays the lattice of congruences of a semigroup
      INTEGER*2 S(8,8),T(600,8,8)
      OPEN(5,FILE='sem.in')
      OPEN(6,FILE='latcong.ans')
      OPEN(7,FILE='chain.dat')
      OPEN(8,FILE='latcong.dat')
    3 FORMAT(1X,I6,I3)
    4 FORMAT(1X,<N>I2)
    5 FORMAT(1X,' ')
    6 FORMAT(1X,'Congruences')
   15 CONTINUE
      WRITE(6,5)
      WRITE(6,5)
      READ(5,3,END=50) ID,N
      WRITE(6,3) ID,N
      DO 20 I=1,N
      READ(5,4) (S(I,J),J=1,N)
      WRITE(6,4) (S(I,J),J=1,N)
   20 CONTINUE
      WRITE(6,5)
      WRITE(6,6)
      CALL GENERATE(T,NT,S,N)
      DO 40 L=1,NT
```

```
      WRITE(6,3) L,N
      DO 30 I=1,N
      WRITE(6,4) (T(L,I,J),J=1,N)
30 CONTINUE
40 CONTINUE
      CALL LATTICE(T,NT,N,KC,ID)
      IF(KC.EQ.0) GO TO 15
      WRITE(7,3) ID,N
      DO 45 I=1,N
      WRITE(7,4) (S(I,J),J=1,N)
45 CONTINUE
      GO TO 15
50 CONTINUE
      STOP
      END

      SUBROUTINE GENERATE(T,NT,S,N)
      INTEGER*2 T(600,8,8),S(8,8),A(8,8)
      NT=0
      DO 10 L=1,N
      DO 10 M=1,N
      A(L,M)=0
10 CONTINUE
      I=N
      J=N
20 CONTINUE
      A(I,J)=A(I,J)+1
      IF(A(I,J).LE.1) GO TO 50
      A(I,J)=-1
      IF(J.EQ.1) GO TO 30
      J=J-1
      GO TO 40
30 CONTINUE
```

```
      J=N
      I=I-1
40 CONTINUE
      IF(I.EQ.0) RETURN
      GO TO 20
50 CONTINUE
      CALL TEST(A,S,N,KT)
      IF(KT.EQ.0) GO TO 20
      IF(I.EQ.N.AND.J.EQ.N) GO TO 70
      IF(J.EQ.N) GO TO 60
      J=J+1
      GO TO 20
60 J=1
      I=I+1
      GO TO 20
70 CONTINUE
      NT=NT+1
      DO 80 L=1,N
      DO 80 M=1,N
      T(NT,L,M)=A(L,M)
80 CONTINUE
      GO TO 20
      END

      SUBROUTINE TEST(A,S,N,KT)
      INTEGER*2 A(8,8),S(8,8)
      KT=0
      I=0
10 I=I+1
      IF(I.GT.N) GO TO 50
      IF(A(I,I).EQ.0) RETURN
      J=0
20 J=J+1
```

```
      IF(J.GT.N) GO TO 10
      IF(A(I,J).EQ.1.AND.A(J,I).EQ.0) RETURN
      IF(A(J,I).EQ.1.AND.A(I,J).EQ.0) RETURN
      K=0
   30 K=K+1
      IF(K.GT.N) GO TO 20
      IF(A(I,J).NE.1.OR.A(J,K).NE.1) GO TO 40
      IF(A(I,K).EQ.0) RETURN
   40 CONTINUE
      IF(A(I,J).NE.1) GO TO 30
      IF(A(S(I,K),S(J,K)).EQ.0) RETURN
      IF(A(S(K,I),S(K,J)).EQ.0) RETURN
      GO TO 30
   50 CONTINUE
      KT=1
      RETURN
      END

      SUBROUTINE LATTICE(T,NT,N,KC,ID)
      INTEGER*2 T(600,8,8),B(600,600)
    3 FORMAT(1X,' ')
    4 FORMAT(1X,'Lattice of Congruences')
    5 FORMAT(1X,<NT>I2)
    6 FORMAT(1X,I6,I3)
      DO 50 L=1,NT
      DO 50 M=1,NT
      B(L,M)=0
      I=0
   10 I=I+1
      IF(I.GT.N) GO TO 30
      J=0
   20 J=J+1
      IF(J.GT.N) GO TO 10
```

```
      IF(T(L,I,J).EQ.0) GO TO 20
      IF(T(M,I,J).EQ.0) GO TO 40
      GO TO 20
30 CONTINUE
      B(L,M)=1
40 CONTINUE
50 CONTINUE
      WRITE(6,3)
      WRITE(6,4)
      WRITE(8,6) ID,NT
      DO 60 L=1,NT
      WRITE(8,5) (B(L,M),M=1,NT)
      WRITE(6,5) (B(L,M),M=1,NT)
60 CONTINUE
      KC=0
      I=0
70 I=I+1
      IF(I.GT.NT) GO TO 90
      J=0
80 J=J+1
      IF(J.LT.I) GO TO 80
      IF(J.GT.NT) GO TO 70
      IF(B(I,J).EQ.0) RETURN
      GO TO 80
90 CONTINUE
      KC=1
      RETURN
      END
```

Example. In this example we display, as does the program, the input semigroup and it lattice of congruences. For the purpose of giving a sample of the output, we display only one of the congruence relation adjacency matrices. It is easily discovered that the congruences do not form a chain for this particular semigroup.

557 5

1	1	1	1	1
1	1	1	1	2
3	3	3	3	3
3	3	3	4	3
1	1	3	3	5

Congruence Number 4

1	0	1	1	0
0	1	0	0	0
1	0	1	1	0
1	0	1	1	0
0	0	0	0	1

Lattice of Congruences

1	1	1	1	1	1	1	1	1	1
0	1	0	1	0	1	0	0	1	1
0	0	1	1	0	0	1	1	1	1
0	0	0	1	0	0	0	0	1	1
0	0	0	0	1	1	1	1	1	1
0	0	0	0	0	1	0	0	1	1
0	0	0	0	0	0	1	1	1	1
0	0	0	0	0	0	0	0	1	1
0	0	0	0	0	0	0	0	0	1

CONGRUENCE EXTENSION PROPERTY
SORTING PROGRAM

A semigroup S has the congruence extension property provided that each congruence σ on each subsemigroup T of S can be extended to a congruence ρ on S,i.e., $\rho \cap (T \times T) = \sigma$. These semigroups have been characterized in [Tang, 1993]. It is occasionally convenient to have a list of these semigroups in hand in order to compare this property to others.

The input file for this program is **sem.in** and contains a list of semigroups in standard format. There are two output files, each in standard format, which separate those semigroups with the congruence extension property (CEP) and those without. The file **cep.dt** contains the list of those semigroups with CEP and the file **nocep.dt** contains the list of those that do not have CEP.

cep.f

```
C Congruence Extension Property Sorting Program
      INTEGER*2 S(8,8),F(256,8),H(256,8,8)
      INTEGER*2 P(4140,8,8),A(4140),B(4140)
      INTEGER*2 D(8)
      OPEN(5,FILE='sem.in')
      OPEN(8,FILE='cep.dt')
      OPEN(9,FILE='nocep.dt')
    3 FORMAT(1X,I6,I3)
    4 FORMAT(1X,¡N¿I2)
      NEW=0
   10 CONTINUE
      IDX=1
      READ(5,3,END=150) ID,N
      LAPFLAG=1
      IF(NEW.EQ.N) GO TO 20
      NEW=N
      CALL SET(N,F,NF)
      CALL RELATION(N,P,NP)
   20 CONTINUE
```

```
      DO 30 I=1,N
      READ(5,4) (S(I,J),J=1,N)
   30 CONTINUE
      CALL SUBSEM(S,N,F,NF,H,NH)
      CALL FLAG(S,N,P,NP,A)
      DO 50 L=1,NP
      IF(A(L).EQ.0) GO TO 50
   50 CONTINUE
      DO 140 L=1,NH
      CALL CONGSUB(H,L,S,N,P,NP,B)
      CALL ELEMENT(H,L,N,NSUB,D)
      CALL SAME(H,L,N,P,NP,B)
      DO 130 M=1,NP
      IF(B(M).EQ.0) GO TO 120
      K=0
   70 K=K+1
      IF(K.GT.NP) GO TO 110
      IF(A(K).EQ.0) GO TO 70
      I=0
   80 I=I+1
      IF(I.GT.N) GO TO 100
      J=0
   90 J=J+1
      IF(J.GT.N) GO TO 80
      IF(H(L,I,J).EQ.0) GO TO 90
      IF(P(M,I,J).NE.P(K,I,J)) GO TO 70
      GO TO 90
  100 CONTINUE
      GO TO 120
  110 CONTINUE
      LAPFLAG=0
      IF(IDX.EQ.0) GO TO 120
      IDX=0
```

```
120 CONTINUE
130 CONTINUE
140 CONTINUE
      IF(LAPFLAG.EQ.0) GO TO 145
      WRITE(8,3) ID,N
      DO 142 ITX=1,N
      WRITE(8,4) (S(ITX,ITY),ITY=1,N)
142 CONTINUE
      GO TO 10
145 CONTINUE
      WRITE(9,3) ID,N
      DO 146 ITX=1,N
      WRITE(9,4) (S(ITX,ITY),ITY=1,N)
146 CONTINUE
      GO TO 10
150 CONTINUE
      STOP
      END

      SUBROUTINE SET(N,F,NF)
      INTEGER*2 F(256,8),G(8)
      NF=0
      DO 10 I=1,N
      G(I)=0
   10 CONTINUE
      J=N
   20 CONTINUE
      G(J)=G(J)+1
      IF(G(J).LE.1) GO TO 30
      G(J)=-1
      J=J-1
      IF(J.EQ.0) RETURN
      GO TO 20
```

```
30 CONTINUE
   IF(J.EQ.N) GO TO 40
   J=J+1
   GO TO 20
40 CONTINUE
   NSUM=0
   DO 45 I=1,N
   NSUM=NSUM+G(I)
45 CONTINUE
   IF(NSUM.EQ.1) GO TO 20
   NF=NF+1
   DO 50 I=1,N
   F(NF,I)=G(I)
50 CONTINUE
   GO TO 20
   END

   SUBROUTINE SUBSEM(S,N,F,NF,H,NH)
   INTEGER*2 S(8,8),F(256,8),H(256,8,8)
   NH=0
   DO 50 K=1,NF
   CALL CLOSE(S,N,F,K,KC)
   IF(KC.EQ.0) GO TO 40
   NH=NH+1
   NUM=0
   DO 10 I=1,N
   NUM=NUM+F(K,I)
10 CONTINUE
   DO 30 I=1,N
   DO 30 J=1,N
   H(NH,I,J)=0
   NPR=F(K,I)*F(K,J)
   IF(NPR.EQ.0) GO TO 30
```

```
      H(NH,I,J)=S(I,J)
   30 CONTINUE
   40 CONTINUE
   50 CONTINUE
      RETURN
      END

      SUBROUTINE CLOSE(S,N,F,K,KC)
      INTEGER*2 S(8,8),F(256,8)
      KC=0
      I=0
   10 I=I+1
      IF(I.GT.N) GO TO 30
      IF(F(K,I).EQ.0) GO TO 10
      J=0
   20 J=J+1
      IF(J.GT.N) GO TO 10
      IF(F(K,J).EQ.0) GO TO 20
      IF(F(K,S(I,J)).EQ.0) RETURN
      GO TO 20
   30 CONTINUE
      KC=1
      RETURN
      END

      SUBROUTINE RELATION(N,P,NP)
      INTEGER*2 A(8,8),P(4140,8,8)
      NM=N-1
      NP=0
      CALL INIT(A,N)
      I=N-1
      J=N
   20 CONTINUE
      A(I,J)=A(I,J)+1
```

```
      IF(A(I,J).LE.1) GO TO 50
      A(I,J)=-1
      IP=I+1
      IF(J.EQ.IP) GO TO 30
      J=J-1
      GO TO 40
   30 CONTINUE
      J=N
      I=I-1
   40 CONTINUE
      IF(I.EQ.0) RETURN
      GO TO 20
   50 CONTINUE
      IF(I.EQ.NM.AND.J.EQ.N) GO TO 70
      IF(J.EQ.N) GO TO 60
      J=J+1
      GO TO 20
   60 CONTINUE
      I=I+1
      J=I+1
      GO TO 20
   70 CONTINUE
      CALL EQUIV(A,N,KC)
      IF(KC.EQ.0) GO TO 20
      NP=NP+1
      CALL TRANSFER(A,N,P,NP)
      GO TO 20
      END

      SUBROUTINE INIT(A,N)
      INTEGER*2 A(8,8)
      NM=N-1
      DO 20 I=1,N
```

```
      DO 10 J=1,N
      A(I,J)=0
10 CONTINUE
      A(I,I)=1
20 CONTINUE
      A(NM,N)=-1
      RETURN
      END

      SUBROUTINE EQUIV(A,N,KC)
      INTEGER*2 A(8,8)
      KC=0
      I=0
10 I=I+1
      IF(I.GT.N) GO TO 40
      J=I-1
20 J=J+1
      IF(J.GT.N) GO TO 10
      K=J-1
30 K=K+1
      IF(K.GT.N) GO TO 20
      NS=A(I,J)+A(J,K)+A(I,K)
      IF(NS.EQ.2) RETURN
      GO TO 30
40 CONTINUE
      KC=1
      RETURN
      END

      SUBROUTINE TRANSFER(A,N,P,NP)
      INTEGER*2 A(8,8),P(4140,8,8)
      I=0
10 I=I+1
      IF(I.GT.N) RETURN
```

```
      J=I-1
   20 J=J+1
      IF(J.GT.N) GO TO 10
      P(NP,I,J)=A(I,J)
      P(NP,J,I)=A(I,J)
      GO TO 20
      END

      SUBROUTINE FLAG(S,N,P,NP,A)
      INTEGER*2 S(8,8),P(4140,8,8),A(4140)
      DO 60 L=1,NP
      A(L)=0
      I=0
   10 I=I+1
      IF(I.GT.N) GO TO 40
      J=0
   20 J=J+1
      IF(J.GT.N) GO TO 10
      IF(P(L,I,J).EQ.0) GO TO 20
      K=0
   30 K=K+1
      IF(K.GT.N) GO TO 20
      IF(P(L,S(I,K),S(J,K)).EQ.0) GO TO 50
      IF(P(L,S(K,I),S(K,J)).EQ.0) GO TO 50
      GO TO 30
   40 CONTINUE
      A(L)=1
   50 CONTINUE
   60 CONTINUE
      RETURN
      END

      SUBROUTINE CONGSUB(H,L,S,N,P,NP,B)
      INTEGER*2 H(256,8,8),S(8,8),P(4140,8,8),B(4140)
```

```
      DO 60 M=1,NP
      B(M)=0
      I=0
   10 I=I+1
      IF(I.GT.N) GO TO 40
      J=0
   20 J=J+1
      IF(J.GT.N) GO TO 10
      IF(H(L,I,J).EQ.0) GO TO 20
      IF(P(M,I,J).EQ.0) GO TO 20
      K=0
   30 K=K+1
      IF(K.GT.N) GO TO 20
      IF(H(L,I,K).EQ.0.OR.H(L,J,K).EQ.0) GO TO 30
      IF(P(M,S(I,K),S(J,K)).EQ.0) GO TO 50
      IF(H(L,K,I).EQ.0.OR.H(L,K,J).EQ.0) GO TO 30
      IF(P(M,S(K,I),S(K,J)).EQ.0) GO TO 50
      GO TO 30
   40 CONTINUE
      B(M)=1
   50 CONTINUE
   60 CONTINUE
      RETURN
      END

      SUBROUTINE SAME(H,L,N,P,NP,B)
      INTEGER*2 H(256,8,8),P(4140,8,8),B(4140),Q(4140)
    3 FORMAT(1X,'Equivalent Subsemigroup Congruences')
    4 FORMAT(1X,¡NUM¿I4)
    5 FORMAT(1X,' ')
      DO 90 M=1,NP
      IF(B(M).EQ.0) GO TO 90
      NQ=1
```

```
      Q(1)=M
      K=0
10    K=K+1
      IF(K.GT.NP) GO TO 50
      IF(K.EQ.M) GO TO 10
      I=0
20    I=I+1
      IF(I.GT.N) GO TO 40
      J=0
30    J=J+1
      IF(J.GT.N) GO TO 20
      IF(H(L,I,J).EQ.0) GO TO 30
      IF(P(M,I,J).NE.P(K,I,J)) GO TO 10
      GO TO 30
40    CONTINUE
      NQ=NQ+1
      Q(NQ)=K
      B(K)=0
      GO TO 10
50    CONTINUE
      MQ=NQ/20
      IF(MQ.EQ.0) GO TO 70
      NUM=20
      LQ=NQ-20*MQ
      KL=1
      KU=20
      DO 60 I=1,MQ
      KL=KL+20
      KU=KU+20
60    CONTINUE
      IF(LQ.EQ.0) GO TO 80
      NUM=LQ
      GO TO 80
```

```
70 CONTINUE
   NUM=NQ
80 CONTINUE
90 CONTINUE
   RETURN
   END

   SUBROUTINE ELEMENT(H,L,N,NSUB,D)
   INTEGER*2 H(256,8,8),D(8)
   NSUB=0
   DO 20 I=1,N
   NS=0
   DO 10 J=1,N
   NS=NS+H(L,I,J)
10 CONTINUE
   IF(NS.EQ.0) GO TO 20
   NSUB=NSUB+1
   D(NSUB)=I
20 CONTINUE
   RETURN
   END
```

In view of the fact that the cep.f is a sorting program, we will not display a sample of the output here, since it would simply be a list of semigroups in standard format with or without CEP.

IDEAL EXTENSION PROPERTY
SORTING PROGRAM

A semigroups S has the ideal extension property (IEP) provided that for each subsemigroup T and each ideal I of T, there exists an ideal J of S such that $J \cap T = I$. Commutative ideal semigroups were characterized in [Aucoin, 1993]. The program listed here iep.f was useful in generating examples both to investigate to obtain generalizations and for counterexamples. The general characterization problem has, to this authors knowledge, not yet been solved. How this property relates to the congruence extension property (CEP) is not transparent. It is known from [Garcia, 1991] that for commutative semigroups having CEP implies having IEP, and it is conjectured that this remains true in the absence of commutativity. There are examples of semigroups (both commutative and non-commutative) with IEP that do not have CEP.

The program calls for an input file **sem.in** which, as with all sorting programs, is a list of semigroups in standard format, and outputs two lists, **iep.dt** (those semigroups with IEP) and **noiep.dt** (those semigroups without IEP).

iep.f

```
C Ideal extension property sorting program
      INTEGER*2 S(8,8),T(256,8),H(256,8)
      OPEN(5,FILE='sem.in')
      OPEN(6,FILE='iep.dt')
      OPEN(7,FILE='noiep.dt')
    3 FORMAT(1X,I6,I3)
    4 FORMAT(1X,¡N¿I2)
   10 CONTINUE
      READ(5,3,END=60) ID,N
      DO 20 I=1,N
      READ(5,4) (S(I,J),J=1,N)
   20 CONTINUE
      CALL GEN(S,N,T,NT,H,NH)
      CALL TEST(S,N,T,NT,H,NH,KP)
```

```
    IF(KP.EQ.0) GO TO 40
    WRITE(6,3) ID,N
    DO 30 I=1,N
    WRITE(6,4) (S(I,J),J=1,N)
30 CONTINUE
    GO TO 10
40 CONTINUE
    WRITE(7,3) ID,N
    DO 50 I=1,N
    WRITE(7,4) (S(I,J),J=1,N)
50 CONTINUE
    GO TO 10
60 CONTINUE
    STOP
    END
    SUBROUTINE GEN(S,N,T,NT,H,NH)
    INTEGER*2 S(8,8),T(256,8),H(256,8),F(8)
    NT=0
    NH=0
    DO 5 L=1,N
    F(L)=0
 5 CONTINUE
    F(N)=-1
    I=N       10 CONTINUE
    F(I)=F(I)+1
    IF(F(I).LE.1) GO TO 20
    F(I)=-1 I=I-1
    IF(I.EQ.0) RETURN
    GO TO 10
20 CONTINUE
    IF(I.EQ.N) GO TO 30
    I=I+1
    GO TO 10
```

```
30 CONTINUE
   MS=0
   DO 35 M=1,N
   MS=MS+F(M)
35 CONTINUE
   IF(MS.EQ.0) GO TO 10
   CALL SUBTEST(S,N,F,KS)
   IF(KS.EQ.0) GO TO 10
   NT=NT+1
   DO 40 L=1,N
   T(NT,L)=F(L)
40 CONTINUE
   CALL IDLTEST(S,N,F,KI)
   IF(KI.EQ.0) GO TO 10
   NH=NH+1
   DO 50 L=1,N
   H(NH,L)=F(L)
50 CONTINUE
   GO TO 10
   END
   SUBROUTINE SUBTEST(S,N,F,KS)
   INTEGER*2 S(8,8),F(8)
   KS=0
   I=0
10 CONTINUE
   I=I+1
   IF(I.GT.N) GO TO 30
   IF(F(I).EQ.0) GO TO 10
   J=0
20 CONTINUE
   J=J+1
   IF(J.GT.N) GO TO 10
   IF(F(J).EQ.0) GO TO 20
```

```
      IF(F(S(I,J)).EQ.0) RETURN
      IF(F(S(J,I)).EQ.0) RETURN
      GO TO 20
   30 CONTINUE
      KS=1
      RETURN
      END
      SUBROUTINE IDLTEST(S,N,F,KI)
      INTEGER*2 S(8,8),F(8)
      KI=0
      I=0
   10 CONTINUE
      I=I+1
      IF(I.GT.N) GO TO 30
      IF(F(I).EQ.0) GO TO 10
      K=F(I)*I
      J=0
   20 CONTINUE
      J=J+1
      IF(J.GT.N) GO TO 10
      IF(F(S(K,J)).EQ.0) RETURN
      IF(F(S(J,K)).EQ.0) RETURN
      GO TO 20
   30 CONTINUE
      KI=1
      RETURN
      END
      SUBROUTINE TEST(S,N,T,NT,H,NH,KP)
      INTEGER*2 S(8,8),T(256,8),H(256,8),A(8),B(8)
      KP=1
      L=0
   10 L=L+1
      IF(L.GT.NT) RETURN
```

```
      DO 20 K=1,N
      B(K)=T(L,K)
      A(K)=0
20 CONTINUE
      A(N)=-1
      I=N
30 CONTINUE
      A(I)=A(I)+1
      IF(A(I).LE.1) GO TO 40
      A(I)=-1
      I=I-1
      IF(I.EQ.0) GO TO 10
      GO TO 30
40 CONTINUE
      IF(I.EQ.N) GO TO 50
      I=I+1
      GO TO 30
50 CONTINUE
      MS=0
      DO 55 M=1,N
      MS=MS+A(M)
55 CONTINUE
      J=0
60 CONTINUE
      J=J+1
      IF(J.GT.N) GO TO 70
      IF(A(J).EQ.1.AND.B(J).EQ.0) GO TO 30
      GO TO 60
70 CONTINUE
      IF(MS.EQ.0) GO TO 30
      CALL CHECK(S,N,H,NH,A,B,KP)
      IF(KP.EQ.0) RETURN
      GO TO 30
```

```
      END
      SUBROUTINE CHECK(S,N,H,NH,A,B,KP)
      INTEGER*2 S(8,8),H(256,8),A(8),B(8)
      KP=1
      I=0
   10 CONTINUE
      I=I+1
      IF(I.GT.N) GO TO 40
      IF(A(I).EQ.0) GO TO 10
      K=A(I)*I
      J=0
   20 CONTINUE
      J=J+1
      IF(J.GT.N) GO TO 10
      IF(B(J).EQ.0) GO TO 20
      M=J*B(J)
      IF(A(S(K,M)).EQ.0) RETURN
      IF(A(S(M,K)).EQ.0) RETURN
      GO TO 20
   40 CONTINUE
      L=0
   50 CONTINUE
      L=L+1
      IF(L.GT.NH) GO TO 70
      I=0
   60 CONTINUE
      I=I+1
      IF(I.GT.N) RETURN
      IF(B(I).EQ.1.AND.H(L,I).NE.A(I)) GO TO 50
      GO TO 60
   70 CONTINUE
      KP=0
      RETURN
```

```
END
```

Again, since this is a sorting program, we will not display a sample of the output.

IDEAL SEMIGROUP
SORTING PROGRAM

A semigroup S is called an ideal semigroup provided each congruence on S is determined by an ideal, i.e., each congruence on S is of the form $I \times I \cup \Delta(S)$, where I is an ideal of S and $\Delta(S)$ is the diagonal relation of S.

The structure of commutative ideal semigroups has been characterized in [Aucoin, 1995]. Again, to this authors knowledge, no general structure theorem exists.

The program **idealsem.f** accepts a list of semigroups as input in standard format and sorts these into two files **idsem.dt** containing those semigroups which are ideal semigroups and **noidsem.dt** containing those semigroups which are not ideal semigroups. There is an additional output file called **extra.ans** which lists the input semigroups, its ideals, and its congruences.

idealsem.f

```
C Ideal semigroup sorting program
      INTEGER*2 S(8,8),F(256,8),P(4140,8,8) INTEGER*2 G(256,8),C(4140,8,8)
      OPEN(5,FILE='sem.in')
      OPEN(7,FILE='idsem.dt')
      OPEN(6,FILE='noidsem.dt')
      OPEN(8,FILE='extra.ans')
    3 FORMAT(1X,I6,I3)
    4 FORMAT(1X,¡N¿I2)
      NEW=0
   10 CONTINUE
      READ(5,3,END=50) ID,N
      IF(NEW.EQ.N) GO TO 20
      CALL SET(N,F,NF)
      CALL RELATION(N,P,NP)
      NEW=N
   20 CONTINUE
      DO 30 I=1,N
      READ(5,4) (S(I,J),J=1,N)
```

```
30 CONTINUE
   CALL IDEALS(S,N,F,NF,G,NG)
   CALL CONGSORT(S,N,P,NP,C,NC)
   CALL MATCH(N,C,NC,G,NG,MAT)
   CALL EXTRA(S,ID,N,C,NC,G,NG)
   M=MAT+6
   WRITE(M,3) ID,N
   DO 40 I=1,N
   WRITE(M,4) (S(I,J),J=1,N)
40 CONTINUE
   GO TO 10
50 CONTINUE
   STOP
   END

   SUBROUTINE SET(N,F,NF)
   INTEGER*2 F(256,8),G(8)
   NF=0
   DO 10 I=1,N
   G(I)=0
10 CONTINUE
   J=N
20 CONTINUE
   G(J)=G(J)+1
   IF(G(J).LE.1) GO TO 30
   G(J)=-1
   J=J-1
   IF(J.EQ.0) RETURN
   GO TO 20
30 CONTINUE
   IF(J.EQ.N) GO TO 40
   J=J+1
   GO TO 20
```

```
40 CONTINUE
   NF=NF+1
   DO 50 I=1,N
   F(NF,I)=G(I)
50 CONTINUE
   GO TO 20
   END

   SUBROUTINE RELATION(N,P,NP)
   INTEGER*2 A(8,8),P(4140,8,8)
   NM=N-1
   NP=0
   CALL INIT(A,N)
   I=N-1
   J=N
20 CONTINUE
   A(I,J)=A(I,J)+1
   IF(A(I,J).LE.1) GO TO 50
   A(I,J)=-1
   IP=I+1
   IF(J.EQ.IP) GO TO 30
   J=J-1
   GO TO 40
30 CONTINUE
   J=N
   I=I-1
40 CONTINUE
   IF(I.EQ.0) RETURN
   GO TO 20
50 CONTINUE
   IF(I.EQ.NM.AND.J.EQ.N) GO TO 70
   IF(J.EQ.N) GO TO 60
   J=J+1
```

```
      GO TO 20
60 CONTINUE
      I=I+1
      J=I+1
      GO TO 20
70 CONTINUE
      CALL EQUIV(A,N,KC)
      IF(KC.EQ.0) GO TO 20
      NP=NP+1
      CALL TRANSFER(A,N,P,NP)
      GO TO 20
      END

      SUBROUTINE INIT(A,N)
      INTEGER*2 A(8,8)
      NM=N-1
      DO 20 I=1,N
      DO 10 J=1,N
      A(I,J)=0
10 CONTINUE
      A(I,I)=1
20 CONTINUE
      A(NM,N)=-1
      RETURN
      END

      SUBROUTINE EQUIV(A,N,KC)
      INTEGER*2 A(8,8)
      KC=0
      I=0
10 I=I+1
      IF(I.GT.N) GO TO 40
      J=I-1
20 J=J+1
```

```
      IF(J.GT.N) GO TO 10
      K=J-1
   30 K=K+1
      IF(K.GT.N) GO TO 20
      NS=A(I,J)+A(J,K)+A(I,K)
      IF(NS.EQ.2) RETURN
      GO TO 30
   40 CONTINUE
      KC=1
      RETURN
      END

      SUBROUTINE TRANSFER(A,N,P,NP)
      INTEGER*2 A(8,8),P(4140,8,8)
      I=0
   10 I=I+1
      IF(I.GT.N) RETURN
      J=I-1
   20 J=J+1
      IF(J.GT.N) GO TO 10
      P(NP,I,J)=A(I,J)
      P(NP,J,I)=A(I,J)
      GO TO 20
      END

      SUBROUTINE IDEALS(S,N,F,NF,G,NG)
      INTEGER*2 S(8,8),F(256,8),G(256,8)
      NG=0
      DO 50 K=1,NF
      I=0
   10 I=I+1
      IF(I.GT.N) GO TO 30
      IF(F(K,I).EQ.0) GO TO 10
      J=0
```

```
20 J=J+1
   IF(J.GT.N) GO TO 10
   IF(F(K,S(I,J)).EQ.0) GO TO 50
   IF(F(K,S(J,I)).EQ.0) GO TO 50
   GO TO 20
30 CONTINUE
   NG=NG+1
   DO 40 M=1,N
   G(NG,M)=F(K,M)
40 CONTINUE
50 CONTINUE
   RETURN
   END

   SUBROUTINE CONGSORT(S,N,P,NP,C,NC)
   INTEGER*2 S(8,8),P(4140,8,8),C(4140,8,8)
   NC=0
   DO 60 K=1,NP
   I=0
10 I=I+1
   IF(I.GT.N) GO TO 40
   J=0
20 J=J+1
   IF(J.GT.N) GO TO 10
   IF(P(K,I,J).EQ.0) GO TO 20
   L=0
30 L=L+1
   IF(L.GT.N) GO TO 20
   IF(P(K,S(I,L),S(J,L)).EQ.0) GO TO 60
   IF(P(K,S(L,I),S(L,J)).EQ.0) GO TO 60
   GO TO 30
40 CONTINUE
   NC=NC+1
```

```
      DO 50 MA=1,N
      DO 50 MB=1,N
      C(NC,MA,MB)=P(K,MA,MB)
50 CONTINUE
60 CONTINUE
      RETURN
      END

      SUBROUTINE MATCH(N,C,NC,G,NG,MAT)
      INTEGER*2 C(4140,8,8),G(256,8),Q(256,8,8)
      MAT=0
      M=0
10 M=M+1
      IF(M.GT.NG) GO TO 40
      I=0
20 I=I+1
      Q(M,I,I)=1
      IF(I.GT.N) GO TO 10
      J=0
30 J=J+1
      IF(J.GT.N) GO TO 20
      IF(I.EQ.J) GO TO 30
      Q(M,I,J)=G(M,I)*G(M,J)
      GO TO 30
40 CONTINUE
      K=0
50 K=K+1
      IF(K.GT.NC) GO TO 60
      GO TO 70
60 CONTINUE
      MAT=1
      RETURN
70 CONTINUE
```

```
      M=0
   80 M=M+1
      IF(M.GT.NG) RETURN
      I=0
   90 I=I+1
      IF(I.GT.N) GO TO 50
      J=0
  100 J=J+1
      IF(J.GT.N) GO TO 90
      IF(C(K,I,J).NE.Q(M,I,J)) GO TO 80
      GO TO 100
      END

      SUBROUTINE EXTRA(S,ID,N,C,NC,G,NG)
      INTEGER*2 S(8,8),C(4140,8,8),G(256,8),A(8)
    2 FORMAT('*')
    3 FORMAT(1X,I6,I3)
    4 FORMAT(1X,¡N¿I2)
    5 FORMAT(1X,' ')
    6 FORMAT(1X,'Ideals')
    7 FORMAT(1X,¡M¿I2)
    8 FORMAT(1X,'Congruences')
    9 FORMAT(1X,'Semigroup')
      WRITE(8,2)
      WRITE(8,5)
      WRITE(8,5)
      WRITE(8,9)
      WRITE(8,5)
      WRITE(8,3) ID,N
      DO 10 I=1,N
      WRITE(8,4) (S(I,J),J=1,N)
   10 CONTINUE
      WRITE(8,5)
```

```
      WRITE(8,6)
      WRITE(8,5)
      DO 40 K=1,NG
      L=0
      M=0
   20 L=L+1
      IF(L.GT.N) GO TO 30
      IF(G(K,L).EQ.0) GO TO 20
      M=M+1
      A(M)=L
      GO TO 20
   30 CONTINUE
      WRITE(8,7) (A(I),I=1,M)
   40 CONTINUE
      WRITE(8,5)
      WRITE(8,8)
      WRITE(8,5)
      DO 60 K=1,NC
      DO 50 I=1,N
      WRITE(8,4) (C(K,I,J),J=1,N)
   50 CONTINUE
      WRITE(8,5)
   60 CONTINUE
      WRITE(8,5)
      RETURN
      END
```

This program computes the semigroup algebra of a semigroup with zero (1 is the designated zero element) and the Lie algebra of this semigroup over the field $\mathbb{Z}_p$ (integers mod p for a prime p). The two parameters are set in the early part of the source program. Comments in the program listing provide a guideline for setting these. The order of the semigroups in the input list N is provided prior to reading the list of input semigroups, so that some computations can be performed that will apply to all of the semigroups in the list (and hence all of the input semigroups must be of the same order N). Setting the parameter P has some limitations due to the magnitude of the computational problems invoked, but the major restriction here is the size of the output file.

If S is a semigroup with zero (again, designated as 1), the semigroup algebra of S over $\mathbb{Z}_p$ consists of the functions f from S into $\mathbb{Z}_p$ such that $f(1) = 0$. The addition of the algebra is defined by $(f + g)(x) = f(x) + g(x)$, where the second $+$ is in $\mathbb{Z}_p$, and is therefore independent of the multiplication in S, as it depends only on p and N. The multiplication in the algebra is defined by $(f * g)(x) = \sum_{ab=x} f(a)g(a)$, the multiplication and sum occurring in the field $\mathbb{Z}_p$; and $(f * g)(x) = 0$ if x is not a product of two elements in S. The Lie Algebra multiplication is defined by $[f, g] = f * g - g * f$.

Input for this program is placed in **sem.in** as a list of semigroups all with zero (1), and all of the same order N in standard format. Output includes a listing of the functions in the semigroup algebra, the addition and multiplication tables, and the Lie Algebra table. These are self-explanatory. Here is the program listing:

alg.f

```
C This program displays the semigroup algebra of input semigroups
      INTEGER*2 P,S(9,9),H(729,9),A(729,729),Q(729)
      OPEN(5,FILE='sem.in')
      OPEN(6,FILE='alg.ans')
    2 FORMAT(1X,' ')
    3 FORMAT(I2)
    5 FORMAT(1X,'Semigroup Algebra over Z(',I1,')')
```

```
    6 FORMAT('*')
      WRITE(6,6)
C Set N at the order of the input semigroups (needed in advance of the input file)
      N=3
C Set P at the order of the finite field Z(P) (P is a positive integer)
C Computational max P**(N-1) LE 729 C Output max P**(N-1) LE 40
      P=3
      KSW=1
      WRITE(6,2)
      WRITE(6,5) P
      WRITE(6,2)
      CALL DEF(H,N,P,M,KSW)
      WRITE(6,2)
      CALL ADD(H,N,A,P,M,Q)
      WRITE(6,2)
   10 CONTINUE
      CALL LOAD(S,N,LAST)
      IF(LAST.EQ.1) STOP
      CALL MULT(H,N,S,P,M,Q,A)
      GO TO 10
      END

      SUBROUTINE LOAD(S,N,LAST)
      INTEGER*2 S(9,9)
    3 FORMAT(1X,I6,I3)
    4 FORMAT(1X,¡N¿I2)
    5 FORMAT(1X,¡N¿I3)
    6 FORMAT('*')
    7 FORMAT(1X,'Semigroup', 1X,I6,5X,'order',1X,I3)
      LAST=0
      READ(5,3,END=20) IDS,NDUM
      WRITE(6,6)
```

```
      WRITE(6,7) IDS,NDUM
      DO 10 I=1,N
      IF(N.LE.9) READ(5,4) (S(I,J),J=1,N)
      IF(N.LE.9) WRITE(6,4) (S(I,J),J=1,N)
      IF(N.GE.10) READ(5,5) (S(I,J),J=1,N)
      IF(N.GE.10) WRITE(6,5) (S(I,J),J=1,N)
   10 CONTINUE
      RETURN
   20 LAST=1
      RETURN
      END

      SUBROUTINE DEF(H,N,P,M,KSW)
      INTEGER*2 P,H(729,9),B(9),E(9)
    2 FORMAT(1X,' ')
    3 FORMAT(1X,I3,5X,9I2)
    4 FORMAT(1X,'Semigroup elements',3X,9I3)
    5 FORMAT(1X,'Field elements ',3X,9I3)
    6 FORMAT(1X,'Functions (Algebra Elements)')
    M=P**(N-1)
      DO 10 J=1,N
      B(J)=0
      H(1,J)=0
   10 CONTINUE
      I=N
      K=1
   20 CONTINUE
      B(I)=B(I)+1
      IF(B(I).LT.P) GO TO 30
      B(I)=0
      I=I-1
      IF(I.EQ.1) GO TO 50
      GO TO 20
```

```
 30 CONTINUE
    K=K+1
    DO 40 L=1,N
    H(K,L)=B(L)
 40 CONTINUE
    I=N
    GO TO 20
 50 CONTINUE
    IF(KSW.EQ.0) GO TO 55
    CALL SWAP(H,B,M,N)
    GO TO 75
 55 B(1)=1
    DO 70 K=1,M
    ISUM=0
    DO 60 J=1,N
    ISUM=ISUM+H(K,J)
    IF(H(K,J).EQ.1) KN=J
 60 CONTINUE
    IF(ISUM.EQ.1) B(KN)=K
 70 CONTINUE
 75 CONTINUE
    DO 76 J=1,N
    E(J)=B(J)-1
 76 CONTINUE
    WRITE(6,5) (E(J),J=1,N)
    WRITE(6,4) (J,J=1,N)
    WRITE(6,2)
    WRITE(6,2)
    WRITE(6,6)
    WRITE(6,2)
    DO 100 I=1,M
    WRITE(6,3) I,(H(I,J),J=1,N)
100 CONTINUE
```

```
      WRITE(6,2)
      RETURN
      END

      SUBROUTINE ADD(H,N,A,P,M,Q)
      INTEGER*2 P,H(729,9),A(729,729),T(9),Q(729)
    2 FORMAT(1X,' ')
    3 FORMAT(1X,'Addition Table')
    4 FORMAT(1X,5X,40I3)
    5 FORMAT(1X,I3,2X,40I3)
    6 FORMAT(1X,'Additive Inverse Table')
    7 FORMAT(1X,40I3)
      DO 50 I=1,M
      DO 50 K=1,M
      DO 10 J=1,N
      T(J)=MOD(H(I,J)+H(K,J),P)
   10 CONTINUE
      LG=0
   20 LG=LG+1
      L=0
   30 L=L+1
      IF(L.GT.N) GO TO 40
      IF(T(L).EQ.H(LG,L)) GO TO 30
      GO TO 20
   40 CONTINUE
      A(I,K)=LG
   50 CONTINUE
      DO 70 I=1,M
      L=0
   60 L=L+1
      IF(A(I,L).NE.1) GO TO 60
      Q(I)=L
   70 CONTINUE
```

```
      WRITE(6,2)
      WRITE(6,6)
      WRITE(6,7) (I,I=1,M)
      WRITE(6,7) (Q(I),I=1,M)
      WRITE(6,2)
      WRITE(6,3)
      WRITE(6,2)
      WRITE(6,4) (NUM,NUM=1,M)
      WRITE(6,2)
      DO 90 I=1,M
      WRITE(6,5) I,(A(I,K),K=1,M)
90 CONTINUE
      RETURN
      END

      SUBROUTINE MULT(H,N,S,P,M,Q,A)
      INTEGER*2 P,H(729,9),S(9,9),T(729,729)
      INTEGER*2 B(9),Q(729),A(729,729)
      INTEGER*2 R(729)
 2 FORMAT(1X,' ')
 3 FORMAT(1X,'Multiplication Table')
 4 FORMAT(1X,'Lie Algebra')
 5 FORMAT(1X,5X,40I3)
 6 FORMAT(1X,I3,2X,40I3)
      DO 60 I=1,M
      DO 60 K=1,M
      DO 10 J=1,N
      B(J)=0
10 CONTINUE
      DO 20 L=1,N
      DO 20 J=1,N
      MQ=S(L,J)
      B(MQ)=MOD(H(I,L)*H(K,J)+B(MQ),P)
```

```
20 CONTINUE
   B(1)=0
   LA=0
30 LA=LA+1
   LB=0
40 LB=LB+1
   IF(LB.GT.N) GO TO 50
   IF(B(LB).EQ.H(LA,LB)) GO TO 40
   GO TO 30
50 CONTINUE
   T(I,K)=LA
60 CONTINUE
   WRITE(6,2)
   WRITE(6,3)
   WRITE(6,2)
   WRITE(6,5) (NUM,NUM=1,M)
   WRITE(6,2)
   DO 65 I=1,M
   WRITE(6,6) I,(T(I,K),K=1,M)
65 CONTINUE
   WRITE(6,2)
   WRITE(6,4)
   WRITE(6,2)
   WRITE(6,5) (NUM,NUM=1,M)
   WRITE(6,2)
   DO 90 I=1,M
   DO 80 K=1,M
   R(K)=A(T(I,K),Q(T(K,I)))
80 CONTINUE
   WRITE(6,6) I,(R(K),K=1,M)
90 CONTINUE
   RETURN
   END
```

```
      SUBROUTINE SWAP(H,B,M,N)
      INTEGER*2 H(729,9),B(9)
    2 FORMAT(1X,' ')
      WRITE(6,2)
      DO 40 I=2,N
      IP=I+1
      DO 40 K=IP,M
      ISUM=0
      DO 10 J=1,N
      ISUM=ISUM+H(K,J)
   10 CONTINUE
      IF(ISUM.GT.1) GO TO 40
      DO 20 J=1,N
      IF(H(K,J).NE.0) KF=J
   20 CONTINUE
      IF(KF.NE.I) GO TO 40
      DO 30 J=1,N
      LT=H(K,J)
      H(K,J)=H(I,J)
      H(I,J)=LT
   30 CONTINUE
   40 CONTINUE
      DO 50 J=1,N
      B(J)=J
   50 CONTINUE
      RETURN
      END
```

Example. We provide here a sample of the output for a low order semigroup, which indicates the magnitude of the restrictions on the order and size of the field $\mathbf{Z}_p$.

Semigroup Algebra over Z(3)

Field elements 0 1 2

Semigroup elements 1 2 3

Functions (algebra elements)

1	0	0	0
2	0	1	0
3	0	0	1
4	0	0	2
5	0	1	1
6	0	1	2
7	0	2	0
8	0	2	1
9	0	2	2

Additive Inverse Table

1	2	3	4	5	6	7	8	9
1	7	4	3	9	8	2	6	5

Addition Table

	1	2	3	4	5	6	7	8	9
1	1	2	3	4	5	6	7	8	9
2	2	7	5	6	8	9	1	3	4
3	3	5	4	1	6	2	8	9	7
4	4	6	1	3	2	5	9	7	8
5	5	8	6	2	9	7	3	4	1
6	6	9	2	5	7	8	4	1	3
7	7	1	8	9	3	4	2	5	6
8	8	3	9	7	4	1	5	6	2
9	9	4	7	8	1	3	6	2	5

Semigroup 11 Order 3

1	1	1
1	2	2
1	3	3

Multiplication Table

	1	2	3	4	5	6	7	8	9
1	1	1	1	1	1	1	1	1	1
2	1	2	2	7	7	1	7	1	2
3	1	3	3	4	4	1	4	1	3
4	1	4	4	3	3	1	3	1	4
5	1	5	5	9	9	1	9	1	5
6	1	6	6	8	8	1	8	1	6
7	1	7	7	2	2	1	2	1	7
8	1	8	8	6	6	1	6	1	8
9	1	9	9	5	5	1	5	1	9

Lie Algebra

	1	2	3	4	5	6	7	8	9
1	1	1	1	1	1	1	1	1	1
2	1	1	6	8	6	8	1	6	8
3	1	8	1	1	8	8	6	6	6
4	1	6	1	1	6	6	8	8	8
5	1	8	6	8	1	6	6	8	1
6	1	6	6	8	8	1	8	1	6
7	1	1	8	6	8	6	1	8	6
8	1	8	8	6	6	1	6	1	8
9	1	6	8	6	1	8	8	6	1

This program determines the actions of a finite semigroup on a finite set X whose cardinal number is specified by setting the parameter M in the early part of the program. A file of input semigroups in input and the actions of these on a set of cardinal number M with elements $1, 2, \ldots, M$ is determined.

Input file is **sem.in** and there is a single output file **act.ans**. This file displays the elements of $\mathrm{Map}(X, X)$, the input semigroup, the homomorphisms of the input semigroup T into $\mathrm{Map}(X, X)$, and a list of these actions in matrix form where the (i, j)-entry is the image of $i \in X$ under the action of $j \in T$.

Each action is specified to be so, if it is divisible, transitive, effective, or unitary.

An action $\phi\colon X \times S \to X$ of a semigroup S on a set X is

1. *divisible* provided that for each $y \in X$, and each $t \in S$, there exists $x \in X$ such that $\phi(x, t) = y$.
2. *transitive* if for each $x, y \in X$, there exists $s \in S$ such that $\phi(x, s) = y$.
3. *effective* if S has an identity e which is the only element of S such that $\phi(x, e) = x$ for all $x \in X$.
4. *unitary* if S has an identity e such that $\phi(x, e) = x$ for all $x \in X$.

act.f

```
C Determines the actions of the
C input semigroup T on a set X
      INTEGER*2 T(8,8),D(256,256),E(256,4)
      INTEGER*2 H(256,8),W(8,8)
      OPEN(5,FILE='sem.in')
      OPEN(6,FILE='act.ans')
    3 FORMAT(1X,I6,I3)
    4 FORMAT(1X,<N>I2)
    5 FORMAT(1X,I2)
    6 FORMAT(1X,'Card X =',I3)
    7 FORMAT(1X,' ')
    8 FORMAT(1X,'Divisible')
    9 FORMAT(1X,'Action',I4)
```

```
  100 FORMAT(1X,'Transitive')
  105 FORMAT(1X,'Unitary')
  108 FORMAT(1X,'Effective')
C The parameter M (the card of X) must be set
      M = 3
      WRITE(6,7)
      WRITE(6,6) M
      WRITE(6,7)
      CALL MAPX(M,D,ND,E)
   10 CONTINUE
      READ(5,3,END=50) ID,N
      WRITE(6,7)
      WRITE(6,3) ID,N
      DO 20 I=1,N
      READ(5,4) (T(I,J),J=1,N)
      WRITE(6,4) (T(I,J),J=1,N)
   20 CONTINUE
      CALL HOM(T,N,D,ND,H,NH)
      DO 45 K=1,NH
      DO 30 I=1,M
      DO 30 J=1,N
      W(I,J)=E(H(K,J),I)
   30 CONTINUE
      CALL DIV(W,N,M,KD)
      WRITE(6,7)
      WRITE(6,9) K
      IF(KD.EQ.1) WRITE(6,8)
      CALL TRANSITIVE(W,N,M,KT)
      IF(KT.EQ.1) WRITE(6,100)
      CALL UNITARY(W,T,N,M,KU)
      IF(KU.EQ.1) WRITE(6,105)
      CALL EFFECTIVE(W,N,M,KE)
      IF(KE.EQ.1) WRITE(6,108)
```

```
      DO 40 I=1,M
      WRITE(6,4) (W(I,J),J=1,N)
   40 CONTINUE
   45 CONTINUE
      GO TO 10
   50 CONTINUE
      STOP
      END

      SUBROUTINE HOM(T,N,D,ND,H,NH)
      INTEGER*2 T(8,8),D(256,256),H(256,8),F(8)
    3 FORMAT(1X,' ')
    4 FORMAT(1X,'Homomorphisms of T into Map(X,X)')
    5 FORMAT(1X,I6,3X,<N>I6)
      WRITE(6,3)
      WRITE(6,4)
      WRITE(6,3)
      NH=0
      DO 10 L=1,N
      F(L)=1
   10 CONTINUE
      K=N
      F(K)=0
   20 CONTINUE
      F(K)=F(K)+1
      IF(F(K).LE.ND) GO TO 30
      F(K)=0
      K=K-1
      IF(K.EQ.0) RETURN
      GO TO 20
   30 CONTINUE
      CALL HOMTEST(T,N,D,F,KF)
      IF(KF.EQ.0) GO TO 20
```

```
      IF(K.EQ.N) GO TO 40
      K=K+1
      GO TO 20
   40 CONTINUE
      NH=NH+1
      DO 50 L=1,N
      H(NH,L)=F(L)
   50 CONTINUE
      WRITE(6,5) NH,(F(L),L=1,N)
      GO TO 20
      END

      SUBROUTINE HOMTEST(T,N,D,F,KF)
      INTEGER*2 T(8,8),D(256,256),F(8)
      KF=0
      I=0
   10 I=I+1
      IF(I.GT.N) GO TO 30
      MA=F(I)
      IF(MA.EQ.0) GO TO 10
      J=0
   20 J=J+1
      IF(J.GT.N) GO TO 10
      MB=F(J)
      IF(MB.EQ.0) GO TO 20
      MC=F(T(I,J))
      IF(MC.EQ.0) GO TO 20
      IF(D(MA,MB).NE.MC) RETURN
      GO TO 20
   30 CONTINUE
      KF=1
      RETURN
      END
```

```
      SUBROUTINE MAPX(M,D,ND,E)
      INTEGER*2 D(256,256),F(4),E(256,4)
    3 FORMAT(1X,' ')
    4 FORMAT(1X,'Elements of Map(X,X)')
    5 FORMAT(1X,I3,3X,<M>I2)
      ND=0
      WRITE(6,3)
      WRITE(6,4)
      WRITE(6,3)
      DO 10 K=1,M
      F(K)=1
   10 CONTINUE
      I=M
      F(I)=0
   20 CONTINUE
      F(I)=F(I)+1
      IF(F(I).LE.M) GO TO 30
      F(I)=0
      I=I-1
      IF(I.EQ.0) GO TO 60
      GO TO 20
   30 CONTINUE
      IF(I.EQ.M) GO TO 40
      I=I+1
      GO TO 20
   40 CONTINUE
      ND=ND+1
      WRITE(6,5) ND,(F(K),K=1,M)
      DO 50 K=1,M
      E(ND,K)=F(K)
   50 CONTINUE
      GO TO 20
   60 CONTINUE
```

```
      JA=0
   70 JA=JA+1
      IF(JA.GT.ND) RETURN
      LA=0
   80 LA=LA+1
      IF(LA.GT.ND) GO TO 70
      IA=0
   90 IA=IA+1
      KA=0
  100 KA=KA+1
      IF(KA.GT.M) GO TO 110
      IF(E(LA,E(JA,KA)).NE.E(IA,KA)) GO TO 90
      GO TO 100
  110 CONTINUE
      D(JA,LA)=IA
      GO TO 80
      END

      SUBROUTINE DIV(W,N,M,KD)
      INTEGER*2 W(8,8)
      KD=1
      K=0
   10 K=K+1
      IF(K.GT.N) RETURN
      J=0
   20 J=J+1
      IF(J.GT.M) GO TO 10
      I=0
   30 I=I+1
      IF(I.GT.M) GO TO 40
      IF(W(I,K).NE.J) GO TO 30
      GO TO 20
   40 CONTINUE
```

```
      KD=0
      RETURN
      END

      SUBROUTINE TRANSITIVE(A,N,M,KT)
      INTEGER*2 A(8,8)
      KT=0
      K=0
   10 K=K+1
      IF(K.GT.M) GO TO 40
      J=0
   20 J=J+1
      IF(J.GT.M) GO TO 10
      I=0
   30 I=I+1
      IF(I.GT.N) RETURN
      IF(A(I,J).NE.K) GO TO 30
      GO TO 20
   40 CONTINUE
      KT=1
      RETURN
      END

      SUBROUTINE UNITARY(A,S,N,M,KU)
      INTEGER*2 A(8,8),S(8,8)
      KU=0
      I=0
   10 I=I+1
      IF(I.GT.N) RETURN
      J=0
   20 J=J+1
      IF(J.GT.N) GO TO 30
      IF(S(I,J).NE.J) GO TO 10
      IF(S(J,I).NE.J) GO TO 10
```

```
    GO TO 20
30 CONTINUE
    K=0
40 K=K+1
    IF(K.GT.M) GO TO 50
    IF(A(I,K).NE.K) RETURN
    GO TO 40
50 CONTINUE
    KU=1
    RETURN
    END

    SUBROUTINE EFFECTIVE(A,N,M,KE)
    INTEGER*2 A(8,8)
    KE=1
    I=0
10 I=I+1
    IF(I.EQ.M) RETURN
    J=I
20 J=J+1
    IF(J.GT.M) GO TO 10
    K=0
30 K=K+1
    IF(K.GT.N) GO TO 40
    IF(A(K,I).NE.A(K,J)) GO TO 20
    GO TO 30
40 CONTINUE
    KE=0
    RETURN
    END
```

EPILOGUE

The method of [Plemmons, 1969] is the largest contribution to the programs above. Without the technique described in that program, generating finite semigroups would be an overwhelming task. Even with this method, the semigroups of order 7 takes a tremendous amount of time on a reasonably fast computer. Probably the most innovative feature of this paper is the "backtracking" method carried over from some facets of Computer Science.

These and other similar programs were used in developing examples of finite semigroups which were used to discover more general theorems and produce counterexample in a number of works (see [Aucoin, 1993], [Aucoin, 1995], [Dumesnil, 1993], [Hildebrant, 1984], and [Hildebrant, 1993]). They were instrumental in the development of semigroups with and without the congruence extension property in [Garcia, 1988], in response to the challenge set forth in [Biró, Kiss, and Pálfy, 1977a] and [Biró, Kiss, and Pálfy, 1977b]. Results in [Garcia, 1991] provided one of the building blocks for the characterization of semigroups with the congruence extension property in [Tang, 1993].

This author has written and checked in excess of 120 programs on finite semigroups. It has been discovered that when a particular project needs to be served, it is best to customize the program to the particular requirements of that project. Of course, the experience of having written this collection of programs is extremely valuable in an undertaking of this sort.

In reviewing the code in some of the programs listed in this handbook, it was realized that more sophisticated techniques have been developed in the interim. However, not seeing a need to modify these programs, we have taken an attitude of "if it ain't broke don't fix it" regarding rewriting the code.

Having mentioned more sophisticated techniques, it does not seem appropriate not to mention at least one of these. In a number of the programs, it desirable to determine which subsets of a semigroup are subsemigroups or ideals. One method of accomplishing this task is to compute the characteristic functions of the subsets (as a string of 0's and 1's) and determine which of these is the characteristic function of a subsemigroup or an ideal. The method of "backtracking" can be employed here, as it consumes less time in computation, than generating a matrix for a semigroup, since all of the entries are either 0 or 1. Programs listed here which require this method use it. This technique was employed in the

programs cep.f, iep.f, and idealsem.f.

[**Aucoin, 1993**]
K.D. Aucoin, **The Congruence Extension Property, The Ideal Extension Property, and Ideal Semigroups**, Dissertation, Louisiana State University, 1993.

[**Aucoin, 1995**]
K.D. Aucoin, *The structure of commutative ideal semigroups*, Semigroup Forum, 50 (1995), 295-300.

[**Biró, Kiss, and Pálfy, 1977a**] B. Biró, E.W. Kiss, and P.R. Pálfy, *On the congruence extension property*, Colloq. Math., 29 (1977), 129-151.

[**Biró, Kiss, and Pálfy, 1977b**] B. Biró, E.W. Kiss, and P.R. Pálfy, *On the congruence extension property*, Semigroup Forum, 15 (1977), 183-184.

[**Dumesnil, 1993**]
J.A. Dumesnil, **The Congruence Extension Property and Related Topics in Semigroups**, Dissertation, Louisiana State University, 1993.

[**Garcia, 1988**]
J.I. Garcia, **The Congruence Extension Property for Algebraic Semigroups**, Dissertation, Louisiana State University, 1988.

[**Garcia, 1991**]
J.I. Garcia, *The congruence extension property for algebraic semigroups*, Semigroup Forum, 43 (1991), 1-18.

[**Hildebrant, 1984**]
J.A. Hildebrant, *The translational degree of a semigroup*, Semigroup Forum, 30 (1984), 331-349.

[Hildebrant, 1993]
J.A. Hildebrant, *Divisible and exponent actions of semigroups*, Semigroup Forum, 46 (1993), 138-145.

[Plemmons, 1969]
R.J. Plemmons, *Construction and analysis of non-equivalent finite semigroups*, Comp. Probs. in Abstract Algebra, (1969), 223-228.

[Tang, 1993]
Xilin Tang, **Semigroups with the Congruence Extension Property**, Dissertation,
Lanzhou University, P. R. China, 1993.