```
)LOAD BBTPNS

SAVED 14:32:01 12/19/83
WSSIZE IS 315696

)VARS

LABOUNDEDNESSTEST  LACLASSES  LACONDITIONS  LACPU  LAPRINT SYSNETS  SYSNVARSANA  SYSNVARSNETS
CC  CG  ES  ID  IS  MS  MO  NN  PN  SQ  SK  ST  TN  TT

)ENS

ABP     ADDNEXT ALPHAS  BETAS   BETWEEN BOUNDED BOUNDEDTEST     BOUNDED1        BOUNDED2        BOUNDED3        BOUNDED4
BREACHSET       CBOUNDOVER      CFCMSPENDANTES  CHECKBOUNDED    CHECKCBOUND     CHECKCFCMS      CHECKCONDITIONS
CHECKTIME       CLASS   CLASSES CLEARCONTEXT    CLEARNET        CONCAT  CONCAT1 CONCAT2 CONTEXTSIZE     CONTINUE
CPATH   DEADTRANSITIONS DEADTST1        DEADTST2        DELETENETS      DROPOLDNET      ERASECONTEXT    ERASENETS
EXEC    FIND    FINDFIRABLE     FSEQ    GAMMAS  GOFROM  HISTORY IFIP83  IND     ING     INICBOUND       INICHECK
INITIME IOTA    LIBNETS LISTENS LIVE    LIVEDIAGNOSIS   LIVETST1        LIVETST2        LIVETST3        LIVETST4
LOADCONTEXT     LOADED  LOADMERGENET     LOADNET MAKECFCMSGRAPH  MAKECLASSES     MAKENET MAR     MODEF   NETDEF  NEXT
NODE    NODES   NODES2  NORMALIZE       ORMSG   OPTIONS PATHO   PATHO3  PAUSEMSG        PDEADLOCKFOR    PLACES  PLDEF
PREDS   PRINTCFCMS      PRINTCLASS      PRINTCLASSES    PRINTDOM        PRINTGRAPH      PRINTMAR        PRINTNETDEF
PRINTNODE       PRINTNODE2      PSAFE   RAMA    REACHSET        READMERGENET    READNET RECALE  REDUC   REDUC2
RENAMENET       RESTAUREOLDNET  SAVECONTEXT     SAVENET SAVEOLDNET      SETBOUNDEDNESSTEST      SETCBOUND
SETCONDITIONS   SETTIME SOWHATNOW        STEP    STEP0   STORECLASS      STOREINITIALCLASS       SUBSTITUTE
TELLSTATES      TELLTIME        TESTTB  TESTTB2 TIME    TIMEOVER        TIR     TOTALDEADLOCKS  TPATH   TRANSDEF
TRANSITIONS     TRI     TS      TSAFE   TS2     UPDATECFCM      UPRED   USUCC   WRITENET        ZUBEK   AL      BE
CT      EL      LN      MA      NL      NR      P       RD      RG      SP      SU      T       TO

LISTENS

[ 0]  ABP
[ 1]  SP0[0,ω]: EI0 → ES0, P0
[ 2]  RSP0[5,6]: ES0 → ES0, P0
[ 3]  RA0[0,1]: ES0, A0 → EI1
[ 4]  SP1[0,ω]: EI1 → ES1, P1
[ 5]  RSP1[5,6]: ES1 → ES1, P1
[ 6]  RA1[0,1]: ES1, A1 → EI0
[ 7]  RP0[0,1]: RI0, P0 → RR0
[ 8]  AP0[0,2]: RR0 → RI1, A0
[ 9]  RP0[0,1]: RI1, P0 → RR0
[10]  RP1[0,1]: RI1, P1 → RR1
[11]  AP1[0,2]: RR1 → RI0, A1
[12]  RP1[0,1]: RI0, P1 → RR1
[13]  LP0[0,1]: P0 →
[14]  LA0[0,1]: A0 →
[15]  LP1[0,1]: P1 →
[16]  LA1[0,1]: A1 →
[17]  M0=EI0,RI0


[0]   P ADDNEXT P
[1]   ⍝ AJOUT NOUVEAU SUIVANT A UN ETAT
```

```
[2]   SG[1 4 ;P;(,SG[2;P;])⍳TR]←P,REC
```

```
[0]   B←ALPHAS P;X
[1]   ⍝ DONNE VECTEUR BORNES INF POUR ETAT P:
[2]   B←0⍴0
[3]   →(0=X←ST[3;P])/0
[4]   B←ID[ST[2;P]+¯1+⍳X]
```

```
[0]   B←BETAS P;X
[1]   ⍝ DONNE VECTEUR BORNES SUP POUR ETAT P:
[2]   B←0⍴0
[3]   →(0=X←ST[3;P])/0
[4]   B←ID[ST[2;P]+X←¯1+⍳X]
```

```
[ 0]  S←E1 BETWEEN E2;E1;E2;X
[ 1]  ⍝ POUR EXHIBER + COURT CHEMIN ENTRE E1 ET E2
[ 2]  X←E2
[ 3]  →(∨/E1∈X)/END
[ 4]  DEB:E1←USUCC E1
[ 5]  E2←UPRED E2
[ 6]  →(E1∈X)/END
[ 7]  →(∧/(E1∈E1),E2∈E2)/NONE
[ 8]  E1←E1
[ 9]  E2←E2
[10]  →DEB
[11]  END:S←(E1∈E2)/E1
[12]  →0
[13]  NONE:S←⍳0
```

```
[ 0]  BOUNDED;P;P;TR;M;M;A;A;B;B;C;C;LΔΔCPU;LΔΔCLASSES
[ 1]  CLEARCONTEXT
[ 2]  INITIME
[ 3]  INICBOUND
[ 4]  P←STEP0
[ 5]  SK←,1
[ 6]  →CHECKCONDITIONS/SUITE
[ 7]  PAUSEMSG
[ 8]  →0
[ 9]  SUITE: ⍝ ENUMERATION DES ETATS DU RESEAU CHARGE.
[10]  →(GOFROM 1)/LAB
[11]  PAUSEMSG
[12]  →0
[13]  LAB:OKMSG
```

```
[0]   K←BOUNDEDTEST NUMBER
[1]   →(2 3 4 =NUMBER)/E2,E3,E4
[2]   E1:→0,K←BOUNDED1
[3]   E2:→0,K←BOUNDED2
[4]   E3:→0,K←BOUNDED3
[5]   E4:→0,K←BOUNDED4
```

```
[0]   K←BOUNDED1;V;W
[1]   ⍝ TEST DITO KARP AND MILLER'S
[2]   W←REDUC V←ST[1;PREDS]
[3]   V←(MS[W;]∧.≤M)/W
[4]   →(K←0=⍴V)/0
[5]   →(K←0=⍴V←(MS[V;]∨.<M)/V)/0
[6]   K←0,P,1↓V
```

```
[ 0]  K←BOUNDED2;V;W;PR;ABC
[ 1]  ⍝ M ≤ M AND M ≠ M AND D = D
[ 2]  W←REDUC V←ST[1;PR←PREDS]
[ 3]  V←(MS[W;]∧.≤M)/W
[ 4]  →(K←0=⍴V)/0
[ 5]  →(K←0=⍴V←(MS[V;]∨.<M)/V)/0
[ 6]  ⍝ V EST ENSEMBLE DE MARQUAGES AVEC M INFERIEURS
[ 7]  ⍝ ABC DE MEME DIM EXISTE PARMI ETATS PREDS AVEC CES MARQUAGES ?
[ 8]  →(K←0=⍴PR←(ST[1;PR]∈V)/PR)/0
[ 9]  →(K←0=⍴PR←(ST[3;PR]=⍴A)/PR)/0
[10]  ⍝ MEME ABC PARMI CES ETATS ?
[11]  ABC←,A,B,C
[12]  →(0=⍴ABC)/END
[13]  W←ST[2;P]∘.+¯1+⍳⍴ABC
[14]  W←((⍴PR),⍴ABC)⍴ID[;W]
[15]  →(K←0=⍴PR←(WA.=ABC)/PR)/0
```

machine

on cours

$E1 \in S^n(E1)$

$E2 \in P^n(E2)$

$1 \Rightarrow$ ens

```
[4]   E3:→0,K←BOUNDED3
[5]   E4:→0,K←BOUNDED4
```

```
[0]   K←BOUNDED1;V;W
[1]   ⍝ TEST DITO KARP AND MILLER'S
[2]   W←REDUC V←ST[1;PREDS]
[3]   V←(MS[W;]∧.≤M)/W
[4]   →(K←0=ρV)/0
[5]   →(K←0=ρV←(MS[V;]∨.<M)/V)/0
[6]   K←0,P,1↑V
```

```
[ 0]  K←BOUNDED2;V;W;PR;ABC
[ 1]  ⍝ M $ M AND M ≠ M AND D = D
[ 2]  W←REDUC V←ST[1;PR←PREDS]
[ 3]  V←(MS[W;]∧.≤M)/W
[ 4]  →(K←0=ρV)/0
[ 5]  →(K←0=ρV←(MS[V;]∨.<M)/V)/0
[ 6]  ⍝ V EST ENSEMBLE DE MARQUAGES AVEC M INFERIEURS
[ 7]  ⍝ ABC DE MEME DIM EXISTE PARMI ETATS PREDS AVEC CES MARQUAGES ?
[ 8]  →(K←0=ρPR←(ST[1;PR]∈V)/PR)/0
[ 9]  →(K←0=ρPR←(ST[3;PR]=ρA)/PR)/0
[10]  ⍝ MEME ABC PARMI CES ETATS ?
[11]  ABC←,A,B,C
[12]  →(0=ρABC)/END
[13]  W←ST[2;P]∘.+¯1+⍳ρABC
[14]  W←((ρPR),ρABC)ρID[,W]
[15]  →(K←0=ρPR←(W∧.=ABC)/PR)/0
[16]  END:K←0,P,1↑PR
```

```
[0]   K←BOUNDED3
[1]   ⍝ AVEC SEUILS
[2]   'TEST D''ARRET NIVEAU 3 NON ENCORE IMPLANTE'
[3]   LΔBOUNDEDTESTLEVEL←2
[4]   K←BOUNDED2
```

```
[0]   K←BOUNDED4
[1]   ⍝
[2]   'TEST D''ARRET NIVEAU 4 NON ENCORE IMPLANTE'
[3]   LΔBOUNDEDTESTLEVEL←2
[4]   K←BOUNDED2
```

```
[0]   R←BREACHSET C;R
[1]   ⍝ CLASSES DEPUIS LESQUELLES C EST ACCESSIBLE (CPEUT ETRE UN ENS.)
[2]   ⍝ UTILISE LES CFCMS
[3]   R←,C
[4]   DEB:R←REDUC R,(CC∈(CC[R]≠0)/CC[R])/⍳ρCC
[5]   R←REDUC R,(∨/[2] SG[1;;]∈R)/⍳(ρSG)[2]
[6]   →(~(ρR)=ρR)/NEXT
[7]   →0
[8]   NEXT:R←R
[9]   →DEB
```

```
[0]   B←CBOUNDOVER LMT;X
[1]   →(~B←(ρST)[2]$LΔCLASSES)/0
[2]   LΔCLASSES←LΔCLASSES+LMT
```

```
[ 0]  P←CFCMSPENDANTES;C;I;E;S
[ 1]  ⍝ DETERMINE CFCMS PENDANTES
[ 2]  P←⍳0
[ 3]  C←C[ΔC←REDUC(CC≠0)/CC]
[ 4]  I←1
[ 5]  DEB:→(I>ρC)/0
[ 6]  E←(CC=C[I])/⍳ρCC
[ 7]  S←(S≠0)/S←,SG[1;E;]
[ 8]  →(∧/S∈E)/LAB
[ 9]  I←I+1
[10]  →DEB
[11]  LAB:P←P,C[I]
[12]  I←I+1
[13]  →DEB
```

```
[0]  K←CHECKBOUNDED;W
[1]  →(K←1↑W←BOUNDEDTEST LABOUNDEDNESSTEST)/0
[2]  'CLASS ',(⍕W[2]),' FAILED BOUNDEDNESS TEST ',(⍕LABOUNDEDNESSTEST),' COMPARE CLASS ',⍕W[3]
[3]  K←SOWHATNOW
```

```
[0]  K←CHECKCBOUND
[1]  →(K←~CBOUNDOVER LACLASSES)/0
[2]  'OVERFLOW NUMBER OF CLASSES'
[3]  TELLSTATES
[4]  K←SOWHATNOW
```

```
[0]  L←CHECKCFCMS P;I
[1]  L←(⍴P)⍴0
[2]  I←1
[3]  DEB:→(I>⍴P)/0
[4]  L[I]←∧/(⍳(⍴TN)[1])∈SG[2;(CC=P[I])/⍳⍴CC;]
[5]  I←I+1
[6]  →DEB
```

```
[0]  K←CHECKCONDITIONS
[1]  ⍝ CHECKS SPECIFIC CONDITIONS IN LACONDITIONS
[2]  K←1
[3]  →(0=⍴LACONDITIONS)/0
[4]  →(K←⍋LACONDITIONS)/0
[5]  'CONDITION ',LACONDITIONS,' VIOLATED'
[6]  K←SOWHATNOW
```

```
[0]  K←CHECKTIME
[1]  →(K←~TIMEOVER LACPU)/0
[2]  'OVERFLOW CPU TIME'
[3]  TELLTIME
[4]  K←SOWHATNOW
```

```
[0]  CLASS N
[1]  ⍝ DITO CLASSES
[2]  CLASSES N
```

```
[ 0]  CLASSES V;I;M;M
[ 1]  ⍝ LIST PARTIEL ETATS
[ 2]  M← 0 0 ⍴''
[ 3]  I←1
[ 4]  →(~'*'∈V)/LAB
[ 5]  V←(('*'=1↑V)/'1'),V,('*'=¯1↑V)/⍳(⍴ST)[2]
[ 6]  V[(V='*')/⍳⍴V]←' '
[ 7]  V←⍋V
[ 8]  V←V[1]+¯1+⍳1+V[2]-V[1]
[ 9]  →DEB
[10]  LAB:V←,⍋V
[11]  V←REDUC V[⍋V]
[12]  DEB:→(I>⍴V)/FIN
[13]  M←PRINTCLASS V[I]
[14]  →(⎕PW$1+(⍴M)[2]+(⍴M)[2])/SUITE
[15]  ' '
[16]  M
[17]  M← 0 0 ⍴''
[18]  SUITE:M←M CONCAT2 ' ' CONCAT2 M
[19]  I←I+1
[20]  →DEB
[21]  FIN:' '
[22]  M
```

```
[0]  CLEARCONTEXT
[1]  TT←TIME
[2]  MS← 0 0 ⍴0
[3]  ST← 4 0 ⍴0
[4]  SG← 4 0 0 ⍴0
[5]  ID←⍳0
[6]  CC←⍳0
[7]  SK←⍳0
[8]  CG← 2 0 0 ⍴0
```

```
[ 7]  V←⍬
[ 8]  V←V[1]+¯1+⍳1+V[2]-V[1]
[ 9]  →DEB
[10]  LAB:V←,⍪V
[11]  V←REDUC V[⍋V]
[12]  DEB:→(I>⍴V)/FIN
[13]  M←PRINTCLASS V[I]
[14]  →(⎕PW≤1+(⍴M)[2]+(⍴N)[2])/SUITE
[15]  ' '
[16]  M
[17]  M← 0 0 ⍴''
[18]  SUITE:M←M CONCAT2 ' ' CONCAT2 N
[19]  I←I+1
[20]  →DEB
[21]  FIN:' '
[22]  M
```

effacer tous autres vars que système s'il est.

```
[0]  CLEARCONTEXT
[1]  TT←TIME
[2]  NS← 0 0 ⍴0
[3]  ST← 4 0 ⍴0
[4]  SG← 4 0 0 ⍴0
[5]  ID←⍳0
[6]  CC←⍳0
[7]  SK←⍳0
[8]  GG← 2 0 0 ⍴0
```

effacer tous vars autres que syst.

```
[0]  CLEARNET
[1]  ES← 2 0 0 ⍴0
[2]  IS← 2 0 1 ⍴0
[3]  NO←0⍴0
[4]  FN← 0 0 ⍴''
[5]  IN← 0 0 ⍴''
[6]  NN←''
```

```
[0]  T←T1 CONCAT T2
[1]  ⍝ CONCATENE TABLEAUX A 3 PLANS SELON LIGNES.
[2]  →((⍴T1)[3]>(⍴T2)[3])/SUP
[3]  →((⍴T1)[3]=(⍴T2)[3])/EQ
[4]  INF:T1←((⍳(⍴T2)[3])∈⍳(⍴T1)[3])\[3] T1
[5]  →EQ
[6]  SUP:T2←((⍳(⍴T1)[3])∈⍳(⍴T2)[3])\[3] T2
[7]  EQ:T←T1,[2] T2
```

```
[0]  T←T1 CONCAT1 T2
[1]  ⍝ CONCATENE MATRICES SELON LIGNES
[2]  T1←(¯2↑ 1 1 ,⍴T1)⍴,T1
[3]  T2←(¯2↑ 1 1 ,⍴T2)⍴,T2
[4]  →((⍴T1)[2]>(⍴T2)[2])/SUP
[5]  →((⍴T1)[2]=(⍴T2)[2])/EQ
[6]  INF:T1←((⍳(⍴T2)[2])∈⍳(⍴T1)[2])\[2] T1
[7]  →EQ
[8]  SUP:T2←((⍳(⍴T1)[2])∈⍳(⍴T2)[2])\[2] T2
[9]  EQ:T←T1,[1] T2
```

```
[0]  T←T1 CONCAT2 T2
[1]  ⍝ CONCATENE MATRICES SELON COLONNES
[2]  T1←(¯2↑ 1 1 ,⍴T1)⍴,T1
[3]  T2←(¯2↑ 1 1 ,⍴T2)⍴,T2
[4]  →((⍴T1)[1]>(⍴T2)[1])/SUP
[5]  →((⍴T1)[1]=(⍴T2)[1])/EQ
[6]  INF:T1←((⍳(⍴T2)[1])∈⍳(⍴T1)[1])\[1] T1
[7]  →EQ
[8]  SUP:T2←((⍳(⍴T1)[1])∈⍳(⍴T2)[1])\[1] T2
[9]  EQ:T←T1,[2] T2
```

Pourquoi fonctionne ?

```
[0]  W←CONTEXTSIZE
[1]  W←(⍴ID)+(×/⍴MS)+(×/⍴ST)+(×/⍴SG)
```

```
[0]  CONTINUE;P;E;TR;M;N;A;Ā;B;B̄;C;C̄;G;LAACPU;LAACLASSES
[1]  ⍝ POURSUIT L'ENUMERATION OU ARRETEE
[2]  INITIME
[3]  INICBOUND
[4]  →(0=⍴SK)/LAB
[5]  →(GOFROM ¯1↑SK)/LAB
```

```
[6]   PAUSEMSG
[7]   →0
[8]   LAB:OKMSG
```

```
[ 0]   S←E1 CPATH E2;E;X
[ 1]   ⍝ EXTRACTS A C-PATH BETWEEN C-SETS E1 AND E2
[ 2]   S←⍳0
[ 3]   E←E1 BETWEEN E2
[ 4]   →(0=ρE)/0
[ 5]   S←1↑(E1∈E)/E1
[ 6]   E←(~E=S)/E
[ 7]   DEB:→((¯1↑S)∈E2)/0
[ 8]   X←1↑(E∈(,SG[1;¯1↑S;]≠0)/,SG[1;¯1↑S;])/E
[ 9]   S←S,X
[10]   E←(~E=X)/E
[11]   →DEB
```

```
[0]   DT←DEADTRANSITIONS
[1]   DT←(~(⍳(ρTN)[1])∈SG[2;;])/⍳(ρTN)[1]
```

```
[ 0]   DEADTST1
[ 1]   B1[0,0]: P1×2, P2 → PT1
[ 2]   E1[5,5]: PT1 → P3, P4
[ 3]   B2[0,0]: P3 → PT2
[ 4]   E2[20,20]: PT2 → P5
[ 5]   B3[0,0]: P4 → PT3
[ 6]   E3[4,4]: PT3 → P6
[ 7]   B4[0,0]: P5, P6 → PT4
[ 8]   E4[3,3]: PT4 → P7, P2
[ 9]   B5[0,0]: P7 → PT5
[10]   E5[2,2]: PT5 → P1
[11]   M0=P1(2),P2,P3
```

```
[ 0]   DEADTST2
[ 1]   ⍝ NON LIVE, PSEUDO LIVE, SOME DEAD TRANSITIONS
[ 2]   B1[0,0]: P1×2, P2 → PT1
[ 3]   E1[5,5]: PT1 → P3, P4
[ 4]   B2[0,0]: P3 → PT2
[ 5]   E2[20,20]: PT2 → P5
[ 6]   B3[0,0]: P4 → PT3
[ 7]   E3[4,4]: PT3 → P6
[ 8]   B4[0,0]: P5, P6 → PT4
[ 9]   E4[3,3]: PT4 → P7, P2
[10]   B5[0,0]: P7 → PT5
[11]   E5[2,2]: PT5 → P1
[12]   V1:P100→P101
[13]   V2:P101→P102
[14]   V3:P102→P100
[15]   D1:P200→P201
[16]   D2:P201→P200
[17]   M0=P1(2),P2,P3,P100
```

```
[0]   DELETENETS L
[1]   ⍝ ERASES DEFINITIONS OF NETS IN L
[2]   ⍝ UPDATES LIBRARY OF NETS
[3]   ⍝ ERASES TABLEAUS OF NETS IN L
[4]   ERASENETS L
[5]   L←NL NR L
[6]   SYSNETS←SYSNETS EL L
[7]   'DEFINITION(S) OF NET(S) ',(LN L),' ERASED'
[8]   L←⎕EX L
```

```
[0]   DROPOLDNET;M
[1]   ⍝ DROPS COPY OF OLD NET O
[2]   M←RD SYSNVARSNETS
[3]   M←⎕EX M,[2]((ρM)[1],1)ρ'O'
```

```
[0]   ERASECONTEXT L;J;M
[1]   ⍝ ERASE CONTEXTS OF NETS IN L
[2]   M←RD SYSNVARSANA
[3]   J←NL NR L
```

```
[17]  MO=P1(2),P2,P3,P100
```

DIRECTORY

```
[0]   DELETENETS L
[1]   ⍝ ERASES DEFINITIONS OF NETS IN L
[2]   ⍝ UPDATES LIBRARY OF NETS
[3]   ⍝ ERASES TABLEAUS OF NETS IN L
[4]   ERASENETS L
[5]   L←NL NR L
[6]   SYS⍙NETS←SYS⍙NETS EL L
[7]   'DEFINITION(S) OF NET(S) ',(LN L),' ERASED'
[8]   L←⎕EX L
```

```
[0]   DROPOLDNET;M
[1]   ⍝ DROPS COPY OF OLD NET O
[2]   M←RD SYS⍙VARSNETS
[3]   M←⎕EX M,[2]((⍴M)[1],1)⍴'O'
```

```
[0]   ERASECONTEXT L;J;M
[1]   ⍝ ERASE CONTEXTS OF NETS IN L
[2]   M←RD SYS⍙VARSANA
[3]   L←NL NR L
[4]   J←⎕EX L
[5]   'CONTEXT(S) OF NET(S) ',(LN J/[1] L),' ERASED'
```

M !

```
[0]   ERASENETS L;J;M
[1]   ⍝ ERASES TABLEAUS OF NETS IN L AND THEIR CONTEXTS
[2]   ERASECONTEXT L
[3]   M←RD SYS⍙VARSNETS
[4]   L←NL NR L
[5]   J←⎕EX L
[6]   'NET(S) ',(LN J/[1] L),' ERASED'
```

M ?

```
[0]   EXEC MAT;I
[1]   I←1
[2]   DEB:→(I>(⍴MAT)[1])/0
[3]   ⍎MAT[I;]
[4]   I←I+1
[5]   →DEB
```

?

```
[ 0]  PMD←FIND;X;W;E;ABC
[ 1]  PMD← 0 0 0
[ 2]  ⍝ EXISTE CLASSE AVEC MEME MARQUAGE ?
[ 3]  X←(MS∧.=M)/⍳(⍴MS)[1]
[ 4]  PMD[2]←1↑X,0
[ 5]  ⍝ EXISTE DOMAINE AVEC MEME NOMBRE DE VARIABLES ?
[ 6]  E←REDUC(ST[3;]=ρA)/ST[2;]
[ 7]  →(0=ρE)/0
[ 8]  ⍝  EXISTE CLASSE AVEC MEME DOMAINE?
[ 9]  PMD[3]←1↑E,0
[10]  ABC←A,B,,C
[11]  →(0=ρABC)/0
[12]  W←E∘.+ 1+⍳ρABC
[13]  W←((ρE),ρABC)⍴ID[,W]
[14]  W←(W∧.=ABC)/E
[15]  PMD[3]←1↑W,0
[16]  →(0=ρW)/0
[17]  ⍝ EXISTE CLASSE AVEC MEME M ET D ?
[18]  E←((⍉ST[1 2 ;])∧.=PMD[2 3])/⍳(ρST)[2]
[19]  PMD[1]←1↑E,0
```

optimisable : comparer d'abord ρA, ρ[illegible]

[12]

```
[ 0]  FINDFIRABLE;KB;X
[ 1]  ⍝ DETERMINE TIRABLES, CALCULE MINS, MAX ET MODIFIE TT
[ 2]  MINS←A
[ 3]  MAX←-1
[ 4]  →(0=ρTT)/0
[ 5]  X←(REDUC TT) ING TT
[ 6]  →(∧/B<0)/LAB
[ 7]  KB←(B$0)/⍳ρB
[ 8]  MAX←⌊/B[KB]
[ 9]  X←X∧(MINS≤MAX)∧∧/[1] C[KB;]$0
[10]  LAB:MINS←X/MINS
```

```
[11]  TT←X/TT


[ 0]  S←E1 FSEQ E2;T
[ 1]  ⍝ SHORTEST FIRING SEQUENCE BETWEEN C-SETS E1 AND E2
[ 2]  E1←⊂E1
[ 3]  E2←⊂E2
[ 4]  →(∨/(E1,E2)>(⍴SG)[2])/ERROR
[ 5]  S←E1 CPATH E2
[ 6]  →(0=⍴S)/NONE
[ 7]  →((⍴S)>1)/LAB
[ 8]  S←⊃S
[ 9]  →0
[10]  LAB:T←TN[TPATH S;]
[11]  S←(⍕1↓S),' ',LN(((⍴T)[1],4)⍴' -> '),[2] T,[2](((⍴T)[1],1)⍴'/'),[2]⍕((⍴T)[1],1)⍴1↓S
[12]  →0
[13]  NONE:'NO SUCH SEQUENCE'
[14]  S←''
[15]  →0
[16]  ERROR:'CLASSE(S) ',(⍕(~(E1,E2)∈⍳(⍴ST)[2])/E1,E2),' UNKNOWN'
[17]  S←''


[0]   C←GAMMAS P;X
[1]   ⍝ DONNE MATRICE CONTRAINTES POUR ETAT P:
[2]   C← 0 0 ⍴0
[3]   →(0=X←ST[3;P])/0
[4]   C←(X,X)⍴ID[(((2×X)+ST[2;P])+¯1)+⍳X×X]


[ 0]  K←GOFROM P
[ 1]  M←MAR P
[ 2]  A←ALPHAS P
[ 3]  B←BETAS P
[ 4]  C←GAMMAS P
[ 5]  FWD: ⍝ STEP FORWARD, FIRES IF ANY, THE FIRST
[ 6]  ⍝ NOT YET FIRED TRANSITION FROM STATE P.
[ 7]  →(~K←CHECKTIME)/0
[ 8]  →(~K←CHECKCBOUND)/0
[ 9]  ⍝ FIND THE FIRST TR:
[10]  TR←NEXT P
[11]  ⍝ CHECKS IF ANY, IF NONE STEP BACK:
[12]  →(0=⍴TR)/BKW
[13]  ⍝ FIRING TR FROM P, FILLS P WITH - INDEX STATE IF
[14]  ⍝ REACHED STATE IS KNOWN, ITS STORAGE INDEX ELSE:
[15]  P←STEP TR
[16]  ⍝ IF STATE REACHED IS NEW, ENUMERATE FROM IT, ELSE
[17]  ⍝ CARRY ON FROM THE CURRENT STATE:
[18]  →(0≤P)/FWD
[19]  SK←SK,P
[20]  P←P
[21]  M←M
[22]  A←A
[23]  B←B
[24]  C←C
[25]  →(~K←CHECKBOUNDED)/0
[26]  →(~K←CHECKCONDITIONS)/0
[27]  →(~K←CHECKTIME)/0
[28]  →(~K←CHECKCBOUND)/0
[29]  →FWD
[30]  BKW: ⍝ STEPS BACK
[31]  ⍝ CALLED WHEN ALL TRANSITIONS FIRABLE FROM
[32]  ⍝ THE CURRENT STATE HAVE BEEN FIRED.
[33]  SK←¯1↓SK
[34]  →(0=⍴SK)/0
[35]  P←¯1↑SK
[36]  M←MAR P
[37]  A←ALPHAS P
[38]  B←BETAS P
[39]  C←GAMMAS P
[40]  →FWD


[ 0]  HISTORY
[ 1]  'VERSION 6'
[ 2]  'ALGO RAPIDE'
[ 3]  'ALGORITHMIQUE CORRECTE, DITO VERSION 5'
[ 4]  'FORMAT ETATS: M, A, B, C'
[ 5]  'SYSTEMES DE CONTRAINTES
```

```
[26]   →(~K←CHECKCONDITIONS)/0
[27]   →(~K←CHECKTIME)/0
[28]   →(~K←CHECKCBOUND)/0
[29]   →FWD
[30]   BKW: ⍝ STEPS BACK
[31]   ⍝ CALLED WHEN ALL TRANSITIONS FIRABLE FROM
[32]   ⍝ THE CURRENT STATE HAVE BEEN FIRED.
[33]   SK←¯1↓SK
[34]   →(0=ρSK)/0
[35]   P←¯1↑SK
[36]   M←MAR P
[37]   A←ALPHAS P
[38]   B←BETAS P
[39]   C←GAMMAS P
[40]   →FWD
```

```
[ 0]   HISTORY
[ 1]   'VERSION 6'
[ 2]   'ALGO RAPIDE'
[ 3]   'ALGORITHMIQUE CORRECTE, DITO VERSION 5'
[ 4]   'FORMAT ETATS: M, A, B, C'
[ 5]   'SYSTEMES DE CONTRAINTES A,B,C REDONDANTS MAIS NORMALISES'
[ 6]   'PAS DE TEST PBORNE, SAVE SI ECHEC'
[ 7]   'AFFICHAGE PARLANT DES RESULTATS'
[ 8]   'LIMITATION POSSIBLE DU TEMPS CPU ET/OU DU NOMBRE D''ETATS'
[ 9]   'DEFINITION RESEAUX ET E/S AMELIOREES'
[10]   '18/4/83: BUGS TIR ET NORMALIZE CORRIGEES'
[11]   '8/6/83: BUGS T ET BE CORRIGEES'
[12]   '?-/6/83: EXTENSIONS: AFFICHAGE PARTIEL GRAPHE OU ENSEMBLE DE CLASSES'
[13]   '?-/6/83: EXTENSIONS: PAUSE POSSIBLE DANS L''ENUMERATION'
[14]   '?-/6/83: CALCUL INCREMENTAL CFCMS POUR TESTS D''A. ET LIVENESS'
[15]   '14/6/83: TIR CORRIGE, BUG APPEL SUBSTITUTE'
[16]   '14/6/83, NOUVELLES FONCTIONS PRINT, DEDICATED '
[17]   '15/6/83: EXTENSION T-BORNE, SANS TESTS D''ARRET'
[18]   '15/6/83: TESTS D''ARRET PBOUNDED LEVELS 1, 2 AND 4'
[19]   '16/6/83: TSAFE, PSAFE USER DEFINED CONDITIONS'
[20]   '16/6/83: MODIF USAGE LACPU ET LACLASSES'
[21]   '17/6/83: NEW CONTEXT MANAGEMENT FUNCTIONS'
[22]   '17/6/83: SEPARATE FIND AND STORE, SEPARATE M AND D AND CLASS SEARCHS'
[23]   '7/10/83: LIVENESS ANALYSIS'
[24]   '7/10/83: CORRECTION BUG RECALE ET PRINTDOM (CAS SYSTEME VIDE)'
[25]   '7/10/83: BUG BOUNDEDTEST2 ET 4 (DIM ABC)'
[26]   '10/10/83: MODIF I/O ET SIMPLIFICATION LIVENESS ANALYSIS (PAS DE GRAPHE DES CFCMS CONSERVE'
[27]   '11/10/83: BUG P, MA ET TO (CAS NOM PLACE = UN SEUL SYMBOLE)'
[28]   '11/10/83: PRINTCFCMS (ET RESTITUTION MAKECFCMS INSIDE)'
[29]   '12/10/83: FONCTIONS PRINTNETDEF ET SAVENETDEF'
[30]   '13/10/83: MODIF NORMALIZE, CAUSE BUG FORME CANONIQUE'
[31]   '13/10/83: PDEADLOCKFOR MODIFIED, BREACHSET INTRODUCED'
[32]   '19/12/83: NOTE: TS2, TIR, AFFICHAGE SYSTEMES NON OK SI '
[33]   'EXISTE TRANSITION(S) SANS PLACE(S) D''ENTREE, NON CORRIGE'
[34]   '; NECESSITE NOTATIONS PARTICULIERES POUR ω-ENABLED TRS.'
```

⟹ on now.

```
[0]    IFIP83
[1]    T1[4,9]: P1, P2×2 → P3, P4 , P5
[2]    T2[0,2]: P4 → P2
[3]    T3[1,3]: P5 → P2
[4]    T4[0,2]: P3 → P3
[5]    T5[0,3]: P3 → P1
[6]    M0=P1(1),P2(2)
```

```
[ 0]   V←B1 IND B2;I;J;K
[ 1]   ⍝ VECTEUR CAR. A DROITE DE B1 DANS B2
[ 2]   V←(ρB2)ρ0
[ 3]   →((0=ρB1)∨0=ρB2)/0
[ 4]   I←ρV
[ 5]   DEB:→(I≤0)/0
[ 6]   K←B2[I]∈B1
[ 7]   →(~K)/LAB
[ 8]   V[I]←1
[ 9]   J←B1⍳B2[I]
[10]   B1←((J-1)↑B1),J↓B1
[11]   LAB:I←I-1
[12]   →DEB
```

TBag × TBag → ℙ

```
[ 0]  V←B1 ING B2;I;K;J
[ 1]  ⍝ VECTEUR CAR. A GAUCHE DE B1 DANS B2
[ 2]  V←(⍴B2)⍴0
[ 3]  →((0=⍴B1)∨0=⍴B2)/0
[ 4]  I←1
[ 5]  DEB:→(I>⍴V)/0
[ 6]  K←B2[I]∈B1
[ 7]  →(~K)/LAB
[ 8]  V[I]←1
[ 9]  J←B1⍳B2[I]
[10]  B1←((J-1)↓B1),J↓B1
[11]  LAB:I←I+1
[12]  →DEB

[0]   INICBOUND
[1]   LΔΔCLASSES←(⍴ST)[2]+LΔCLASSES

[0]   INICHECK
[1]   INITIME
[2]   LΔΔCPU←LΔCPU
[3]   LΔΔSTATES←LΔSTATES

[0]   INITIME
[1]   LΔΔCPU←(TIME-TT)+LΔCPU

[0]   V←M IOTA N;D
[1]   ⍝ INDEX DES LIGNES DE N DANS M OU ZERO
[2]   D←(⍴M)[1]
[3]   V←1+D+((⍳D)-1+D)⌊.×M∧.=⍉N

[0]   LIBNETS
[1]   ,(((⍴SYSΔNETS)[2]⍴1),0)\[2] SYSΔNETS

[ 0]  LISTFNS;I;M;J
[ 1]  ⎕PW←120
[ 2]  J←' ABCDEFGHIJKLMNOPQRSTUVWXYZΔABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789'
[ 3]  J←TRI(⍴⎕NL 3)⍴J⍳(,⎕NL 3)
[ 4]  I←1
[ 5]  DEB:→(I>⍴J)/0
[ 6]  M←⎕CR(⎕NL 3)[J[I];]
[ 7]  ' '
[ 8]  ' '
[ 9]  (((⍴M)[1],1)⍴'['),[2](0 1 +⍷⍳(1,(⍴M)[1])⍴¯1+⍳(⍴M)[1]),[2](((⍴M)[1],4)⍴'] '),[2] M
[10]  I←I+1
[11]  →DEB

[ 0]  LIVE;P;L
[ 1]  ⍝ TROUVER DEADLOCKS TOTAUX
[ 2]  P←TOTALDEADLOCKS
[ 3]  →(0≠⍴P)/NO
[ 4]  ⍝ TROUVER CFCMS PENDANTES
[ 5]  P←CFCMSPENDANTES
[ 6]  ⍝ TESTER PRESENCE DE TOUTES LES
[ 7]  ⍝ TRANSITIONS DANS CHAQUE CFCM PENDANTE
[ 8]  L←CHECKCFCMS P
[ 9]  →(~∧/L)/NO
[10]  'LIVE',((⍴P)>1)/'S; ',(⍕⍴P),' PENDING CFCMS'
[11]  →CY
[12]  NO:'NOT LIVE (USE LIVEDIAGNOSIS FOR MORE INFORMATION)'
[13]  CY:((~∧/CC=1)/'NOT '),'CYCLIC'

[ 0]  LIVEDIAGNOSIS;TD;DT;X;P;DTT;E;LT
[ 1]  ' '
[ 2]  'LIVENESS DIAGNOSIS:'
[ 3]  '(USE PRINTCFCMS FOR MORE ABOUT CFCMS)'
[ 4]  ' '
[ 5]  ⍝ PRINTS LIVENESS DIAGNOSIS
```

```
[ 0]   LIVE;P;L
[ 1]   ⍝ TROUVER DEADLOCKS TOTAUX
[ 2]   P←TOTALDEADLOCKS
[ 3]   →(0≠⍴P)/NO
[ 4]   ⍝ TROUVER CFCMS PENDANTES
[ 5]   P←CFCMSPENDANTES
[ 6]   ⍝ TESTER PRESENCE DE TOUTES LES
[ 7]   ⍝ TRANSITIONS DANS CHAQUE CFCM PENDANTE
[ 8]   L←CHECKCFCMS P
[ 9]   →(~∧/L)/NO
[10]   'LIVE',((⍴P)>1)/'S: ',(⍕⍴P),' PENDING CFCMS'
[11]   →CY
[12]   NO:'NOT LIVE (USE LIVEDIAGNOSIS FOR MORE INFORMATION)'
[13]   CY:((~∧/CC=1)/'NOT '),'CYCLIC'


[ 0]   LIVEDIAGNOSIS;TD;DT;X;P;DTT;E;LT
[ 1]   ' '
[ 2]   'LIVENESS DIAGNOSIS:'
[ 3]   '(USE PRINTCFCMS FOR MORE ABOUT CFCMS)'
[ 4]   ' '
[ 5]   ⍝ PRINTS LIVENESS DIAGNOSIS
[ 6]   TD←TOTALDEADLOCKS
[ 7]   →(0=⍴TD)/SUITE
[ 8]   X←(⍴TD)>1
[ 9]   'CLASS',(X/'ES'),' ',(⍕TD),(X/' ARE'),((~X)/' IS A'),' TOTAL DEADLOCK',X/'S'
[10]   ' '
[11]   SUITE:DT←DEADTRANSITIONS
[12]   ⍝ DTT← TRANSITIONS QUASI VIVANTES
[13]   DTT←(~DTT∈DT)/DTT←⍳(⍴TN)[1]
[14]   →(0≠⍴TD)/END
[15]   X←CHECKCFCMS P←CFCMSPENDANTES
[16]   P←(X=0)/P
[17]   X←1
[18]   LT←DTT
[19]   DEB:→(X>⍴P)/END0
[20]   LT←(LT∈SG[2;(CC=P[X])/⍳⍴CC;])/LT
[21]   X←X+1
[22]   →DEB
[23]   END0:DTT←(~DTT∈LT)/DTT
[24]   X←1<⍴LT
[25]   →(0=⍴LT)/END
[26]   'TRANSITION',(X/'S '),(LN TN[LT;]),(X/' ARE '),((~X)/' IS '),'LIVE'
[27]   ' '
[28]   END:X←1<⍴DT
[29]   →(0=⍴DT)/END2
[30]   'TRANSITION',(X/'S '),(LN TN[DT;]),(X/' ARE '),((~X)/' IS '),'DEAD FROM THE INITIAL CLASS'
[31]   END2: ⍝ DTT CONTIENT TRANSITIONS NON VIVANTES MAIS QUASIVIVANTES
[32]   X←1
[33]   DEB2:→(X>⍴DTT)/0
[34]   E←PDEADLOCKFOR DTT[X]
[35]   'TRANSITION ',TN[DTT[X];],' IS DEAD FROM CLASS',((1<⍴E)/'ES'),' ',⍕E
[36]   X←X+1
[37]   →DEB2


[0]   LIVETST1
[1]   ⍝ LIVE, 2 CFCMS
[2]   T1:E→D
[3]   T2:E→E
[4]   T3:D,F→B,H,G
[5]   T4:E,F→B,H,I
[6]   T5:D,G,H→D,F
[7]   T6:E,I,H→E,F
[8]   M0=B,G,I,F


[ 0]   LIVETST2
[ 1]   ⍝LIVE, 3 CFCMS
[ 2]   T1:A→B
[ 3]   T2:A→C
[ 4]   T3:A→D
[ 5]   T4:B,F,G→B,E
[ 6]   T5:C,F,H→C,E
[ 7]   T6:D,F,I→D,E
[ 8]   T7:B,E→F,G,A
[ 9]   T8:C,E→F,H,A
```

```
[10]   T9:D,E→F,I,A
[11]   M0=A,E,G,H,I


[0]   LIVETST3
[1]   ⍝ LIVE, 2 CFCMS
[2]   T11:B1→D1
[3]   T21:B1→E1
[4]   T31:D1,F1→B1,H1,G1
[5]   T41:E1,F1→B1,H1,I1
[6]   T51:D1,G1,H1→D1,F1
[7]   T61:E1,I1,H1→E1,F1
[8]   M0=B1,G1,I1,F1


[ 0]   LIVETST4
[ 1]   T11:B1→D1
[ 2]   T21:B1→E1
[ 3]   T31:D1,F1→B1,H1,G1
[ 4]   T41:E1,F1→B1,H1,I1
[ 5]   T51:D1,H1,G1→D1,F1
[ 6]   T61:E1,H1,I1→E1,F1
[ 7]   T1:A→B
[ 8]   T2:A→C
[ 9]   T3:A→D
[10]   T4:B,F,G→B,E
[11]   T5:C,F,H→C,E
[12]   T6:D,F,I→D,E
[13]   T7:B,E→A,F,G
[14]   T8:C,E→A,F,H
[15]   T9:D,E→A,F,I
[16]   M0=B1,F1,G1,I1,A,G,E,H,I


[0]   LOADCONTEXT NOM;MAT
[1]   ⍝ CHARGE CONTEXTE RESEAU NOM
[2]   LOADNET NOM
[3]   MAT←SYSΔVARSANA,[2](ρSYSΔVARSANA)[1]ρ'←'
[4]   MAT←MAT,[2](RD SYSΔVARSANA),[2]((ρSYSΔVARSANA)[1],ρNOM)ρNOM←NR NOM
[5]   EXEC MAT
[6]   'CONTEXT OF ',NN,' LOADED'


[0]   LOADED
[1]   NN,' LOADED'
[2]   (⍕(ρES)[3]),' PLACES, ',(⍕(ρES)[2]),' TRANSITIONS'


[0]   LOADMERGENET NOM
[1]   'FUNCTION LOADMERGENET NOT YET IMPLEMENTED'


[0]   LOADNET NOM;MAT
[1]   CLEARCONTEXT
[2]   MAT←SYSΔVARSNETS,[2](ρSYSΔVARSNETS)[1]ρ'←'
[3]   MAT←MAT,[2](RD SYSΔVARSNETS),[2]((ρSYSΔVARSNETS)[1],ρNOM)ρNOM←NR NOM
[4]   EXEC MAT
[5]   NN←NOM
[6]   NOM,' LOADED,'
[7]   (⍕(ρPN)[1]),' PLACES, ',(⍕(ρTN)[1]),' TRANSITIONS.'


[0]   CG←MAKECFCMSGRAPH;CC
[1]   ⍝ MAKES GRAPH OF CFCMS
[2]   ⍝ USES MAKECLASSES FROM BBAUTOMA PACKAGE
[3]   CC←CC+(CC=0)×⍳ρCC
[4]   CG←(CC MAKECLASSES SG[2 1 ;;])[2 1 ;;]


[ 0]   M←C MAKECLASSES TG;V;T;S;J;N;T
[ 1]   ⍝ C: VECTEUR DES CLASSES.
[ 2]   ⍝ RC: LABELS NON TRAITES SI EXISTENT.
[ 3]   M←TG
[ 4]   →(∧/V←C=⍳ρC)/0
[ 5]   ⍝ CONVERSION INDEX ETATS:
[ 6]   ⎕IO←0
[ 7]   M[1;;]←(0,C)[M[1;;]]
[ 8]   M[1;;]←(0,V\1+⍳+/V)[M[1;;]]
```

```
[0]   LOADNET NOM;MAT
[1]   CLEARCONTEXT
[2]   MAT←SYSΔVARSNETS,[2](ρSYSΔVARSNETS)[1]ρ'←'
[3]   MAT←MAT,[2](RD SYSΔVARSNETS),[2]((ρSYSΔVARSNETS)[1],ρNOM)ρNOM←NR NOM
[4]   EXEC MAT
[5]   NN←NOM
[6]   NOM,' LOADED;'
[7]   (⍴(ρPN)[1]),' PLACES, ',(⍴(ρTN)[1]),' TRANSITIONS.'


[0]   CG←MAKECPCMSGRAPH;CC
[1]   ⍝ MAKES GRAPH OF CPCMS
[2]   ⍝ USES MAKECLASSES FROM BBAUTOMA PACKAGE
[3]   CC←CC+(CC=0)×⍳ρCC
[4]   CG←(CC MAKECLASSES SG[2 1 ;;])[2 1 ;;]


[ 0]  M←C MAKECLASSES TG;V;T;S;J;N;T
[ 1]  ⍝ C: VECTEUR DES CLASSES.
[ 2]  ⍝ RQ: LABELS NON TRAITES SI EXISTENT.
[ 3]  M←TG
[ 4]  →(∧/V←C=⍳ρC)/0
[ 5]  ⍝ CONVERSION INDEX ETATS:
[ 6]  ⎕IO←0
[ 7]  M[1;;]←(0,C)[M[1;;]]
[ 8]  M[1;;]←(0,V\1+⍳+/V)[M[1;;]]
[ 9]  ⎕IO←1
[10]  ⍝
[11]  T←S← 0 0 ρ0
[12]  J←⍳ρC
[13]  DEB:→(0=ρJ)/FIN
[14]  V←(C=1↑J)/⍳ρC
[15]  J←(~J∈V)/J
[16]  N←M[;V;]
[17]  N←(2,×/1↓ρN)ρ,N
[18]  T←N[1;]≠0
[19]  N←T/[2] N
[20]  T←(N[1;]≠0)/N[1;]
[21]  V←(⍳ρT)∈T⍳T
[22]  T←T CONCAT1 V/N[1;]
[23]  S←S CONCAT1 V/N[2;]
[24]  →DEB
[25]  FIN:M←(2,ρT)ρ(,T),,S


[ 0]  K←MAKENET NOM;M;I;L
[ 1]  K←1
[ 2]  M← 1 0 ↓⎕CR NOM
[ 3]  I←1
[ 4]  DEB:→(I>(ρM)[1])/END
[ 5]  L←M[I;]
[ 6]  I←I+1
[ 7]  L←(L≠' ')/L
[ 8]  →('⍝'=1↑L,'⍝')/DEB
[ 9]  →('→='∈L)/TRANS,MAR
[10]  'LINE: ',L,' REJECTED'
[11]  →DEB
[12]  TRANS:⍞'(T''',(''')EE(P''|:' SU ''')TO P''|→' SU L),''''
[13]  →DEB
[14]  MAR:⍞'MA''',(3↓L),''''
[15]  →DEB
[16]  END:→(I>1)/0
[17]  K←0


[0]   M←MAR P
[1]   ⍝ P: LISTE DE POINTEURS D'ETATS
[2]   M←,MS[ST[1;P];]


[0]   V←MODEF;J;K
[1]   ⍝ BUILDS DEFINITION OF M0
[2]   V←(M0≠0)/⍳ρM0
[3]   J←PN[V;]
[4]   K←((ρV),1)ρM0[V]
[5]   V←(K[;1]=1)/⍳(ρK)[1]
[6]   K←(((ρK)[1],1)ρ'('),[2](⍕K),[2]((ρK)[1],1)ρ')'
[7]   K[V;]←' '
[8]   V←J,[2] K,[2]((ρK)[1],1)ρ','
[9]   V←'M0← ',(V≠' ')/V← 1↓,V
[ ]   ⍎((ρW)ρ'⍎'),⍳ρW
```

```
[0]   N←NETDEF;I
[1]   ⍝ BUILDS DEFINITION OF NETLOADED
[2]   N←NN
[3]   I←1
[4]   DEB:→(I>(⍴TN)[1])/END
[5]   N←N CONCAT1 TRANSDEF I
[6]   I←I+1
[7]   →DEB
[8]   END:N←N CONCAT1 NODEF


[0]   T←NEXT P
[1]   ⍝ GIVES FIRST TRANSITION NOT YET FIRED FROM
[2]   ⍝ STATE P (MAY BE EMPTY):
[3]   T←(,(SG[1;P;]=0)∧SG[2;P;]≠0)/,SG[2;P;]
[4]   →(0=⍴T)/0
[5]   T←1↑T


[0]   NODE N
[1]   ⍝ DITO NODES
[2]   NODES N


[ 0]  NODES V;I
[ 1]  ⍝ LIST PARTIEL ETATS
[ 2]  I←1
[ 3]  →(~'*'∈V)/LAB
[ 4]  V←(('*'=1↑V)/'1'),V,('*'=¯1↑V)/⍉(⍴ST)[2]
[ 5]  V[(V='*')/⍳⍴V]←' '
[ 6]  V←⍷V
[ 7]  V←V[1]+¯1+⍳1↓V[2]-V[1]
[ 8]  →DEB
[ 9]  LAB:V←,⍷V
[10]  V←REDUC V[⍋V]
[11]  DEB:→(I>⍴V)/0
[12]  PRINTNODE V[I]
[13]  I←I+1
[14]  →DEB


[ 0]  NODES2 V;I
[ 1]  ⍝ LIST PARTIEL ETATS
[ 2]  I←1
[ 3]  →(~'*'∈V)/LAB
[ 4]  V←(('*'=1↑V)/'1'),V,('*'=¯1↑V)/⍉(⍴CG)[2]
[ 5]  V[(V='*')/⍳⍴V]←' '
[ 6]  V←⍷V
[ 7]  V←V[1]+¯1+⍳1↓V[2]-V[1]
[ 8]  →DEB
[ 9]  LAB:V←,⍷V
[10]  V←REDUC V[⍋V]
[11]  DEB:→(I>⍴V)/0
[12]  PRINTNODE2 V[I]
[13]  I←I+1
[14]  →DEB


[ 0]  NORMALIZE;A;B;C;XB
[ 1]  ⍝ 13/10/83
[ 2]  ⍝ CALCULE MINS ET MAXS EFFECTIFS
[ 3]  XB←(B≠0)/⍳⍴B
[ 4]  →(0=⍴XB)/END
[ 5]  DEB:A←⍷A
[ 6]  B←⍷B
[ 7]  C←⍷C
[ 8]  SUBSTITUTE XB
[ 9]  →(∨/A≠⍷A)/DEB
[10]  →(∨/B≠⍷B)/DEB
[11]  →(∨/,C≠⍷C)/DEB
[12]  END:C←C[⍷B∘.-A
[13]  XB←(B<0)/⍳⍴B
[14]  C[XB;]←B[XB]∘.-A
[15]  C←(¯1+⍳⍴B)⌽[2] C
[16]  C[;1]←B-A
[17]  C←(-¯1+⍳⍴B)⌽[2] C
```

Pour CPCMs.

```
PRINTNODE2 V[I]
I←I+1
→DEB
```

```
NORMALIZE;A;B;C;XB
⍝ 13/10/83
⍝ CALCULE MINS ET MAXS EFFECTIFS
XB←(B≤0)/⍳⍴B
→(0=⍴XB)/END
DEB:A←A
B←B
C←C
SUBSTITUTE XB
→(∨/A≠A)/DEB
→(∨/B≠B)/DEB
→(∨/,C≠C)/DEB
END:C←C[B∘.-A
XB←(B<0)/⍳⍴B
C[XB;]←B[XB]∘.-A
C←(¯1+⍳⍴B)⌽[2] C
C[;1]←B-A
C←(-¯1+⍳⍴B)⌽[2] C
```

si chgt alg → DEB.

```
OKMSG
'BOUNDED'
(⍕(⍴ST)[2]),' STATE CLASSES'
TELLTIME
```

```
OPTIONS
SETTIME
SETCBOUND
SETBOUNDEDNESSTEST
SETCONDITIONS
```

```
PATHO
A:P1,P3→P3,P2
B:P3,P2→P2,P4
C:P2,P4→P4,P1,P5
D:P4,P5→P3
M0=P2(1),P4(1),P5(1)
```

```
PATHO3
A:P1×2,P2,P3→P3×3,P4
B:P1,P2×2,P4→P3,P4×3
C:P3×3,P4×3→P1×3,P2×3
M0=P1(2),P2(2),P3(2),P4(2)
```

```
PAUSEMSG
'ENUMERATION SUSPENDED, ENTER CONTINUE FOR MORE'
(⍕(⍴ST)[2]),' CLASSES ENUMERATED SO FAR.'
TELLTIME
```

```
E←PDEADLOCKFOR T;E
⍝ COMPUTES PARTIAL DEADLOCK SET FOR TR. T
⍝ I.E. CLASSES FROM WHICH (OR FROM THE SUCC. OF) T IS NOT FIRABLE.
⍝ E: CLASSES DEPUIS LESQUELLES T EST TIRABLE
E←(∨/[2] SG[2;;]=T)/⍳(⍴SG)[2]
⍝ E ← BACKWARD REACHSET OF E
E←BREACHSET E
⍝ E ← SUCCESSORS OF E NON IN E
E←REDUC(,SG[1;E;]≠0)/,SG[1;E;]
E←(~E∊E)/E
```

```
PLACES
⎕PW CT FL PN
```

PLDEF : → ABSPLACE
inverse : → REPPLACE

```
P←PN PLDEF W
P←((⍴W)⍴'×'),[2]⍕W
```

```
[2]   P[(W[;1]=1)/⍳(⍴W)[1];]←' '
[3]   P←¯1↓,PN,[2] P,[2](⍴W)⍴','
[4]   P←(P≠' ')/P
```

```
[0]   P←PREDS;X
[1]   ⍝ CALCULE PREDECESSEURS DU SOMMET COURANT
[2]   X←CC[SK]=0
[3]   P←(X/SK),(CC∈CC[(~X)/SK])/⍳⍴CC
```

```
[ 0]  PRINTCFCMS;CG;I;K;X;CC
[ 1]  ⍝ PRINTS GRAPH OF CFCMS AND CFCMS
[ 2]  ' '
[ 3]  'GRAPH OF CFCMS:'
[ 4]  CG←MAKECFCMSGRAPH
[ 5]  NODES2⍪⍳(⍴CG)[2]
[ 6]  ' '
[ 7]  'CONTENTS OF CFCMS:'
[ 8]  CC←REDUC((CC=0)×⍳⍴CC)+CC
[ 9]  I←1
[10]  DEB:→(I>⍴CC)/0
[11]  K←CC[I]
[12]  →(0=CC[K])/LAB
[13]  K←(CC=CC[I])/⍳⍴CC
[14]  LAB:X←(⍴,K)>1
[15]  'G',(⍕I),' CONTAINS CLASS',(X/'ES'),' ',⍕,K
[16]  I←I+1
[17]  →DEB
```

Séparer en PRTCFCMS, CFCM ; , CFCMS ; PRINTCFCMGRAPH, CFCNODES ; (C

```
[0]   M←PRINTCLASS I;J
[1]   ⍝ I EST INDEX CLASSE
[2]   M←'CLASS C',((' '≠J)/J←⍕I),':'
[3]   M←M CONCAT1 'M = ' CONCAT2(L⌊PRINT-4) PRINTMAR I
[4]   M←M CONCAT1 'I : ' CONCAT2(L⌊PRINT-4) PRINTDOM I
```

```
[0]   T←PRINTCLASSES
[1]   ' '
[2]   'LIST OF STATE CLASSES:'
[3]   CLASSES⍪⍳(⍴ST)[2]
```

```
[ 0]  M←F PRINTDOM I;J;K;L;A;B;C;T;V;NAM;W
[ 1]  ⍝ AFFICHAGE DOMAINE
[ 2]  A←ALPHAS I
[ 3]  B←BETAS I
[ 4]  C←GAMMAS I
[ 5]  T←TS2 MAR I
[ 6]  M← 0 0 ⍴''
[ 7]  →(0=⍴A)/0
[ 8]  ⍝ CALCUL INSTANTIATIONS
[ 9]  V←+/[1] T∘.=REDUC T
[10]  W←⍪(⍳J←⌈/V)∘.≤V
[11]  W←(W≠0)/W←,W×V←(⍴W)⍴⍳J
[12]  NAM←((J,1)⍴'.'),[2] RG⍪(J,1)⍴⍳J
[13]  NAM←' ' CONCAT1 1 0 ↓NAM
[14]  NAM←RG(RD TN[T;]),[2] RG NAM[W;]
[15]  ⍝ CONTRAINTES SIMPLES
[16]  J←1
[17]  DEB:→(J>⍴T)/DEB2
[18]  L←NR(⍕A[J]),' ≤ ',NAM[J;],(¯1≠B[J])/' ≤ ',⍕B[J]
[19]  M←M CONCAT1 F CT L
[20]  J←J+1
[21]  →DEB
[22]  DEB2: ⍝ CONTRAINTES COMPLEXES
[23]  K←,⍪(⍴C)⍴⍳⍴A
[24]  K←K CONCAT1,(⍴C)⍴⍳⍴A
[25]  V←~(B[K[1;]]<0)∨(B[K[1;]]-A[K[2;]])≤,C
[26]  K←V/[2] K,[1],C
[27]  J←1
[28]  DEB3:→(J>(⍴K)[2])/0
[29]  L←NR NAM[K[1;J];],' - ',NAM[K[2;J];],' ≤ ',⍕K[3;J]
[30]  M←M CONCAT1 F CT L
[31]  J←J+1
[32]  →DEB3
```

ajouter extension o
tq négatives

modif : (ss w-enabled

```
[ 8]   ⍝ CALCUL INSTANTIATIONS
[ 9]   V←+/[1] T∘.=REDUC T
[10]   W←⍳(⍳J←⌈/V)∘.≤V
[11]   W←(W≠0)/W←,W×V←(ρW)ρ⍳J
[12]   NAM←((J,1)ρ'.'),[2] RG⍕(J,1)ρ⍳J
[13]   NAM←' ' CONCAT1 1 0 ↓NAM
[14]   NAM←RG(RD TN[T;]),[2] RG NAM[W;]
[15]   ⍝ CONTRAINTES SIMPLES
[16]   J←1
[17]   DEB:→(J>ρT)/DEB2
[18]   L←NR(⍕A[J]),' ≤ ',NAM[J;],(¯1≠B[J])/' ≤ ',⍕B[J]
[19]   N←N CONCAT1 F CT L
[20]   J←J+1
[21]   →DEB
[22]   DEB2: ⍝ CONTRAINTES COMPLEXES
[23]   K←,⍉(ρC)ρ⍳ρA
[24]   K←K CONCAT1,(ρC)ρ⍳ρA
[25]   V←~(B[K[1;]]<0)∨(B[K[1;]]-A[K[2;]])≤,C
[26]   K←V/[2] K,[1],C
[27]   J←1
[28]   DEB3:→(J>(ρK)[2])/0
[29]   L←NR NAM[K[1;J];],'- ',NAM[K[2;J];],' ≤ ',⍕K[3;J]
[30]   N←N CONCAT1 F CT L
[31]   J←J+1
[32]   →DEB3
```

```
[0]    PRINTGRAPH
[1]    ' '
[2]    'GRAPH OF STATE CLASSES:'
[3]    NODES⍕⍳(ρST)[2]
```

```
[0]    V←F PRINTMAR J;K
[1]    ⍝ AFFICHE MARQUAGE J
[2]    J←MAR J
[3]    V←RD PN[(J≠0)/⍳ρJ;]
[4]    J←(J≠0)/J
[5]    K←(((ρJ),1)ρ'('),[2] RG⍕((ρJ),1)ρJ
[6]    K←(RD K),[2]((ρJ),1)ρ')'
[7]    V←V,[2] RG((J>1)\[1](J>1)/[1] K),[2]((ρJ),2)ρ', '
[8]    V←F CT ¯1↓NR,RG V
```

```
[0]    PRINTNETDEF
[1]    ⍝ PRINTS DEFINITION OF NET LOADED
[2]    'NET LOADED IS: '
[3]    ' '
[4]    NETDEF
```

```
[ 0]   N←PRINTNODE I;REC;T;M;L;J
[ 1]   N← 0 0 ρ''
[ 2]   →(I>1)/DEB
[ 3]   →(0=ρTS M0)/DEB
[ 4]   REC←⌊/,IS[1;TS M0;]
[ 5]   →(REC≤0)/DEB
[ 6]   N←N CONCAT1 '→ (ρ=',(⍕REC),')/C1'
[ 7]   DEB:→(I>(ρSG)[2])/0
[ 8]   L←⍳0
[ 9]   J←1
[10]   DEB1:→(J>(ρSG)[3])/END1
[11]   →(0=SG[2;I;J])/END1
[12]   L←L,',"',(NR TN[SG[2;I;J];]),'"∈[',(⍕SG[3;I;J]),',∞'
[13]   L←L,((ST[4;I]≠0)/⍕ST[4;I]),((ST[4;I]<0)/'ω'),']'
[14]   L←L,')',((SG[4;I;J]≠0)/'/(ρ=',(⍕SG[4;I;J]),')'),'/?'
[15]   →(0=SG[1;I;J])/E1
[16]   L←(¯1↓L),'C',⍕SG[1;I;J]
[17]   E1:J←J+1
[18]   →DEB1
[19]   END1:N←N CONCAT1 ⎕PW CT('C',(⍕I),'→'),1↓L
```

```
[ 0]   N←PRINTNODE2 I;T;M;L;J
[ 1]   N← 0 0 ρ''
[ 2]   →(I>(ρCG)[2])/0
[ 3]   L←⍳0
[ 4]   J←1
[ 5]   DEB1:→(J>(ρCG)[3])/END1
[ 6]   →(0=CG[2;I;J])/END1
```

mod: (low w-enabled)

pour CFCMS
à revoir pour affich Ci → (T1,

```
[ 7]   L←L,', ',(NR TN[CG[2;I;J];]),'/G',⍕CG[1;I;J]
[ 8]   J←J+1
[ 9]   →DEB1
[10]   END1:N←('G',(⍕I),' → ') CONCAT2(⎕PW-4+⍴⍕I) CT,2↓L
```

```
[0]   K←PSAFE
[1]   ⍝ CHECKS PSAFENESS
[2]   →(K←∧/M≤1)/0
[3]   'CLASS ',(⍕P),' FAILED PSAFE CONDITION'
```

```
[ 0]   RAMA
[ 1]   B1[0,0]: P1, P2 → PT1
[ 2]   E1[5,5]: PT1 → P3, P4
[ 3]   B2[0,0]: P3 → PT2
[ 4]   E2[20,20]: PT2 → P5
[ 5]   B3[0,0]: P4 → PT3
[ 6]   E3[4,4]: PT3 → P6
[ 7]   B4[0,0]: P5, P6 → PT4
[ 8]   E4[3,3]: PT4 → P7, P2
[ 9]   B5[0,0]: P7 → PT5
[10]   E5[2,2]: PT5 → P1
[11]   M0=P1,P2,P3
```

```
[0]   R←REACHSET C;R
[1]   ⍝ CLASSES ACCESSIBLES DEPUIS C
[2]   ⍝ UTILISE LES CFCMS
[3]   R←,C
[4]   DEB:R←REDUC R,(CC∈(CC[R]≠0)/CC[R])/⍳⍴CC
[5]   R←REDUC R,((,SG[1;R;])≠0)/,SG[1;R;]
[6]   →(~(⍴R)=⍴R)/NEXT
[7]   →0
[8]   NEXT:R←R
[9]   →DEB
```

```
[ 0]   READMERGENET NOM
[ 1]   ⍝ READS DEF OF NET NOM AND MERGES WITH NET LOADED
[ 2]   SAVEOLDNET
[ 3]   CLEARCONTEXT
[ 4]   →(~MAKENET NOM)/END
[ 5]   SYSΔNETS←SYSΔNETS AL NOM
[ 6]   NOM,' MERGED WITH ',NN,' UNDER NAME: ',NN,NOM
[ 7]   (⍕(⍴PN)[1]),' PLACES, ',(⍕(⍴TN)[1]),' TRANSITIONS.'
[ 8]   'DEFINITION OF NET ',NN,NOM,' CAN BE SAVED USING SAVENETDEF'
[ 9]   DROPOLDNET
[10]   NN←NN,NOM
[11]   →0
[12]   END:'DEFINITION ABORTED'
[13]   RESTAUREOLDNET
```

WRITENET ?

```
[ 0]   READNET NOM
[ 1]   CLEARCONTEXT
[ 2]   CLEARNET
[ 3]   →(~MAKENET NOM)/END
[ 4]   NOM,' LOADED,'
[ 5]   NN←NOM
[ 6]   SYSΔNETS←SYSΔNETS AL NOM
[ 7]   (⍕(⍴PN)[1]),' PLACES, ',(⍕(⍴TN)[1]),' TRANSITIONS.'
[ 8]   →0
[ 9]   END:'DEFINITION ABORTED'
[10]   CLEARNET
```

```
[ 0]   RECALE;XB
[ 1]   ⍝ CALCULE REC, MODIFIE A, B ET C
[ 2]   REC←0
[ 3]   →(0=⍴A)/0
[ 4]   REC←⌊/A
[ 5]   →(REC≤0)/0
[ 6]   A←A-REC
[ 7]   XB←(B≠0)/⍳⍴B
[ 8]   →(0=⍴XB)/END
```

```
[ 0]  RECALE;XB
[ 1]  ⍝ CALCULE REC, MODIFIE A, B ET C
[ 2]  REC←0
[ 3]  →(0=⍴A)/0
[ 4]  REC←⌊/A
[ 5]  →(REC≤0)/0
[ 6]  A←A-REC
[ 7]  XB←(B$0)/⍳⍴B
[ 8]  →(0=⍴XB)/END
[ 9]  B[XB]←B[XB]-REC
[10]  END:XB←(B<0)/⍳⍴B
[11]  C[XB;]←B[XB]∘.-A


[0]   V←REDUC E
[1]   V←((⍳⍴E)∈E,⍳E)/E


[0]   R←REDUC2 L
[1]   ⍝ REDUCTION MATRICE L
[2]   R←((⍳(⍴L)[1])∈L IOTA L)/[1] L


[0]   RENAMENET NOM
[1]   ⍝ RENAMES LOADED NET
[2]   NN,' RENAMED ',NOM
[3]   NN←NOM


[0]   RESTAUREOLDNET;MAT
[1]   MAT←SYSΔVARSNETS,[2](⍴SYSΔVARSNETS)[1]⍴'←'
[2]   MAT←MAT,[2](RD SYSΔVARSNETS),[2]((⍴SYSΔVARSNETS)[1],1)⍴'0'
[3]   EXEC MAT
[4]   NN,' RESTAURED'


[0]   SAVECONTEXT;MAT
[1]   SAVENET
[2]   MAT←(RD SYSΔVARSANA),[2]((⍴SYSΔVARSANA)[1],⍴NN)⍴NN
[3]   MAT←MAT,[2](((⍴MAT)[1],1)⍴'←'),[2] SYSΔVARSANA
[4]   EXEC MAT
[5]   'CONTEXT OF ',NN,' SAVED'


[0]   SAVENET;MAT
[1]   MAT←(RD SYSΔVARSNETS),[2]((⍴SYSΔVARSNETS)[1],⍴NN)⍴NN
[2]   MAT←MAT,[2](((⍴MAT)[1],1)⍴'←'),[2] SYSΔVARSNETS
[3]   EXEC MAT
[4]   NN,' SAVED'
[5]   SAVENETDEF


[0]   SAVEOLDNET;MAT
[1]   ⍝ SAVE TEMPORARILY LOADED NET
[2]   MAT←(RD SYSΔVARSNETS),[2]((⍴SYSΔVARSNETS)[1],1)⍴'0'
[3]   MAT←MAT,[2](((⍴MAT)[1],1)⍴'←'),[2] SYSΔVARSNETS
[4]   EXEC MAT


[0]   SETBOUNDEDNESSTEST;V
[1]   ⍝ SETS LΔBOUNDEDNESSTEST
[2]   V←'BOUNDEDNESS TEST IS ',(⍕LΔBOUNDEDNESSTEST),'; ENTER NEW VALUE (1≤V≤4) OR RETURN: '
[3]   V←(⍴V)↑⎕,0/⎕←V
[4]   →(0=⍴V)/0


[0]   SETCBOUND;V
[1]   ⍝ SETS LΔCLASSES
[2]   V←'NUMBER OF CLASSES INCREMENT IS: ',(⍕LΔCLASSES),'; ENTER NEW VALUE OR RETURN: '
[3]   V←(⍴V)↑⎕,0/⎕←V
[4]   →(0=⍴V)/0
[5]   LΔCLASSES←1↑±V
```

```
[5]   ⍙BOUNDEDNESSTEST←1↑⍵V

[0]   SETCBOUND;V
[1]   ⍝ SETS ⍙CLASSES
[2]   V←'NUMBER OF CLASSES INCREMENT IS: ',(⍕⍙CLASSES),'; ENTER NEW VALUE OR RETURN: '
[3]   V←(⍴V)↓⎕,0/⎕←V
[4]   →(0=⍴V)/0
[5]   ⍙CLASSES←1↑⍵V

[ 0]  SETCONDITIONS;V
[ 1]  ⍝ SETS ⍙CONDITIONS
[ 2]  'SPECIFIC CONDITION IS ',((0=⍴⍙CONDITIONS)/'EMPTY'),⍙CONDITIONS
[ 3]  'TSAFE, PSAFE AND EMPTY ARE KNOWN CONDITIONS'
[ 4]  V←'ENTER NEW CONDITION OR RETURN: '
[ 5]  V←(⍴V)↓⎕,0/⎕←V
[ 6]  →(0=⍴V)/0
[ 7]  →((5=⍴V)∧∧/(5↑V)='EMPTY')/E1
[ 8]  ⍙CONDITIONS←V
[ 9]  →0
[10]  E1:⍙CONDITIONS←''

[0]   SETTIME;V
[1]   ⍝ SETS ⍙CPU
[2]   V←'CPU TIME INCREMENT IS ',(⍕⍙CPU),'; ENTER NEW VALUE OR RETURN: '
[3]   V←(⍴V)↓⎕,0/⎕←V
[4]   →(0=⍴V)/0
[5]   ⍙CPU←1↑⍵V

[0]   B←SOWHATNOW;X
[1]   X←29↓⎕,⎕←'CONTINUE (C) OR PAUSE (P) ?: '
[2]   →(B←'C'=1↑X)/0

[ 0]  P←STEP TR;TT;REC;PMD;MINS;MAX
[ 1]  ⍝ TIR: CALCULE M, A, B, C ET TT
[ 2]  TIR
[ 3]  ⍝ RECALE: CALCULE REC, MODIFIE A, B ET C
[ 4]  RECALE
[ 5]  ⍝ RECHERCHE CLASSE EGALE
[ 6]  PMD←FIND
[ 7]  →(PMD[1]≠0)/OLD
[ 8]  ⍝ CALCUL MINS, MAX, MODIF TT
[ 9]  FINDFIRABLE
[10]  ⍝ STOCKAGE NOUVELLE CLASSE
[11]  P←STORECLASS 1↓PMD
[12]  P ADDNEXT P
[13]  →END
[14]  OLD:P ADDNEXT PMD[1]
[15]  P←-PMD[1]
[16]  END:P UPDATECFCM P

[ 0]  P←STEP0;MINS;MAX;TT;REC
[ 1]  ⍝ VALEURS PAR DEFAUT ET CALCUL TT
[ 2]  A←0⍴0
[ 3]  B←0⍴0
[ 4]  C← 0 0⍴0
[ 5]  M←M0
[ 6]  MINS←⍳0
[ 7]  MAX←¯1
[ 8]  TT←TS2 M
[ 9]  →(0=⍴TT)/STO
[10]  ⍝ CALCUL A,B,C INITIAUX ET RECALAGE
[11]  A←,IS[1;TT;]
[12]  B←,IS[2;TT;]
[13]  C←B∘.-A
[14]  ⍝ RECALAGE, CALCULE REC
[15]  RECALE
[16]  ⍝ CALCUL TIRABLES, MINS, MAX, TT
[17]  FINDFIRABLE
[18]  STO: ⍝ STOCKAGE ETAT INITIAL
[19]  P←STOREINITIALCLASS
[20]  CC←,0
```

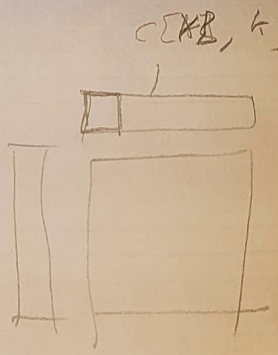
```
[15]  P←-1MD[1]
[16]  END:P UPDATECFCM P
```

```
[ 0]  P←STEP0;MINS;MAX;TT;REC
[ 1]  ⍝ VALEURS PAR DEFAUT ET CALCUL TT
[ 2]  A←0⍴0
[ 3]  B←0⍴0
[ 4]  C← 0 0 ⍴0
[ 5]  M←M0
[ 6]  MINS←⍳0
[ 7]  MAX←¯1
[ 8]  TT←TS2 M
[ 9]  →(0=⍴TT)/STO
[10]  ⍝ CALCUL A,B,C INITIAUX ET RECALAGE
[11]  A←,IS[1;TT;]
[12]  B←,IS[2;TT;]
[13]  C←B∘.-A
[14]  ⍝ RECALAGE, CALCULE REC
[15]  RECALE
[16]  ⍝ CALCUL TIRABLES, MINS, MAX, TT
[17]  FINDTIRABLE
[18]  STO: ⍝ STOCKAGE ETAT INITIAL
[19]  P←STOREINITIALCLASS
[20]  CC←,0
```

```
[ 0]  P←STORECLASS MD
[ 1]  ⍝ STORAGE CLASS
[ 2]  →(0≠MD[1])/E1
[ 3]  MS←MS,[1] M
[ 4]  MD[1]←(⍴MS)[1]
[ 5]  E1:→(0≠MD[2])/E2
[ 6]  MD[2]←1+⍴ID
[ 7]  ID←ID,A,B,,C
[ 8]  E2:ST←ST,[2] MD,(⍴A),MAX
[ 9]  SG←SG CONCAT(4 1 ,(⍴TT))⍴((⍴TT)⍴0),TT,MINS,(⍴TT)⍴0
[10]  P←(⍴ST)[2]
```

```
[0]   P←STOREINITIALCLASS
[1]   ST← 4 1 ⍴ 1 1 ,(⍴A),MAX
[2]   MS←(1,⍴M)⍴M
[3]   SG←(4 1 ,⍴TT)⍴((⍴TT)⍴0),TT,MINS,(⍴TT)⍴0
[4]   ID←A,B,,C
[5]   P←1
```

```
[ 0]  SUBSTITUTE V;K;XB
[ 1]  ⍝ SUBSTITUE VARS D'INDEX V DANS A,B ET C
[ 2]  ⍝ CONSERVE DIMENSIONS A, B ET C
[ 3]  XB←(B⍳0)/⍳⍴B
[ 4]  V←(V∈XB)/V
[ 5]  DEB:→(0=⍴V)/0
[ 6]  K←1↑V
[ 7]  A←A⌈A[K]-,C[K;]
[ 8]  B←B⌊B[K]+,C[;K]
[ 9]  C[XB;]←C[XB;]⌊(,C[XB;K])∘.+,C[K;]
[10]  V←1↓V
[11]  →DEB
```

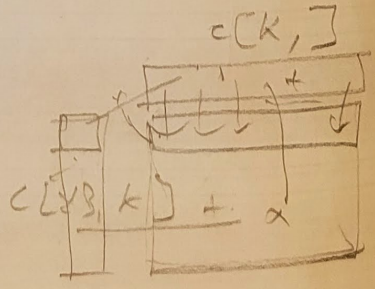



```
[0]   TELLSTATES
[1]   (⍕(⍴ST)[2]),' CLASSES ENUMERATED'
```

```
[0]   TELLTIME
[1]   'CPUT = ',(⍕TIME-TT),'S'
```

```
[0]   TESTTB
[1]   A[2,4]:P1 → P2
[2]   B[3,6]:P2→
[3]   M0←P1(10)
```

```
[0]   TESTTB2
[1]   A[2,4]: → P
[2]   B[2,6]: P →
```



```
[5]   LABOUNDEDNESSTEST←1↓⍵V
[0]   D←TIME
[1]   D←⎕AI[2]-1000

[0]   B←TIMEOVER LMT;X
[1]   →(~B←(TIME-TT)$LΔΔCPU)/0
[2]   LΔΔCPU←LΔΔCPU+LMT

[ 0]  TIR;TSR;TSE;TT;X;XB;K;SUP
[ 1]  ⍝ REMPLIT M,A,B,C,TT ; TIRE TR DEPUIS ETAT P
[ 2]  ⍝ CALCUL NOUVEAU MARQUAGE ET TESTS
[ 3]  M←(M-,ES[1;TR;])+,ES[2;TR;]
[ 4]  A←0ρ0
[ 5]  B←0ρ0
[ 6]  C← 0 0 ρ0
[ 7]  ⍝ CALCUL NOUVELLEMENT SENSIBILISEES
[ 8]  TT←TS2 M
[ 9]  →(0=ρTT)/0
[10]  ⍝ CALCUL RESTANTES ET TEST
[11]  TSR←TS2 M-,ES[1;TR;]
[12]  →(0=ρTSR)/E2
[13]  TT←TS2 M
[14]  ⍝ DECALAGE
[15]  E3:XB←(B$0)/⍳ρB
[16]  K←TT⍳TR
[17]  SUP←ST[4;P]
[18]  A←0⌈(SUP$0)×((B[K]$0)×,-C[K;])⌈A-SUP
[19]  B←B
[20]  C←C
[21]  →(0=ρXB)/E4
[22]  B[XB]←(,C[XB;K])⌊B[XB]-A[K]
[23]  C[XB;]←C[XB;]⌊B[XB]∘.-A
[24]  →((ρXB)=ρA)/LAB2
[25]  C[X;]←((ρA)ρ⍳⍴1)[X←(B<0)/⍳ρB]∘.-A
[26]  LAB2:X←(ρC)ρ1,(ρA)ρ0
[27]  C[XB;]←C[XB;]+X[XB;]×(B[XB]∘.-A)-C[XB;]
[28]  ⍝ ELIMINATION SORTANTES
[29]  E4:SUBSTITUTE(~X←TSR IND TT)/⍳ρTT
[30]  A←X/A
[31]  B←X/B
[32]  C←X/[1] X/[2] C
[33]  ⍝ NORMALISATION
[34]  NORMALIZE
[35]  ⍝ ADJONCTION ENTRANTES
[36]  E2:X←TSR ING TT
[37]  A←X\A
[38]  B←X\B
[39]  C←X\[1] X\[2] C
[40]  TSE←(~X)/TT
[41]  X←(~X)/⍳ρX
[42]  A[X]←,IS[1;TSE;]
[43]  B[X]←,IS[2;TSE;]
[44]  C[X;]←B[X]∘.-A
[45]  C[;X]←B∘.-A[X]
[46]  ⍝ CHECK POINT

[0]   TD←TOTALDEADLOCKS
[1]   TD←(∧/[2] SG[2;;]=0)/⍳(ρSG)[2]

[0]   F←TPATH C
[1]   ⍝ FIRING SEQUENCE FOR C-PATH C
[2]   F←(,SG[1;⍳1↓C;]=⍉(∘.≤ρSG[1;⍳1↓S;])ρ1↓S)/,SG[2;⍳1↓S;]

[0]   T←TRANSDEF I
[1]   ⍝ BUILDS DEFINITION OF TRANSITION I OF NET LOADED
[2]   T←'[',(⍕IS[1;I;]),',',((IS[2;I;]<0)/'ω'),((IS[2;I;]$0)/⍕IS[2;I;]),']:'
[3]   T←TN[I;],((IS[1;I;]≠0)∨IS[2;I;]≠⍳1)/T
[4]   T←(T≠' ')/T
[5]   T←T,': ',((ES[1;I;]≠0)/[1] PN) PLDEF⍉(ES[1;I;]≠0)/[2] ES[1;,I;]
[6]   T←T,' → ',((ES[2;I;]≠0)/[1] PN) PLDEF⍉(ES[2;I;]≠0)/[2] ES[2;,I;]

[0]   TRANSITIONS
```

corriger (cas w-enabled)

ie tq w-enabled sont tjrs
et jamais

↳ dvar à droite de TSR dans TT

↳ dvar à gauche de TSR dans

$\Rightarrow \begin{cases} X \leftarrow d0 \\ X \leftarrow X \times TT \geqslant 0 \end{cases}$

WRITETRANS

```
[42]  A[X]←,IS[1;TSE;]
[43]  B[X]←,IS[2;TSE;]
[44]  C[X;]←B[X]∘.-A
[45]  C[;X]←B∘.-A[X]
[46]  ⍝ CHECK POINT
```

```
[0]   TD←TOTALDEADLOCKS
[1]   TD←(∧/[2] SG[2;;]=0)/⍳(⍴SG)[2]
```

```
[0]   F←TPATH C
[1]   ⍝ FIRING SEQUENCE FOR C-PATH C
[2]   F←(,SG[1;¯1↓C;]=⍉(⍉⍴SG[1;¯1↓S;])⍴1↓S)/,SG[2;¯1↓S;]
```

```
[0]   T←TRANSDEF I
[1]   ⍝ BUILDS DEFINITION OF TRANSITION I OF NET LOADED
[2]   T←'[',(▼IS[1;I;]),',',((IS[2;I;]<0)/'ω'),((IS[2;I;]≠0)/▼IS[2;I;]),']:'
[3]   T←T,⍕IN[I;],((IS[1;I;]≠0)∨IS[2;I;]≠¯1)/T
[4]   T←(T≠' ')/T
[5]   T←T,': ',((ES[1;I;]≠0)/[1] PN) PLDEF⍉(ES[1;I;]≠0)/[2] ES[1;,I;]
[6]   T←T,' → ',((ES[2;I;]≠0)/[1] PN) PLDEF⍉(ES[2;I;]≠0)/[2] ES[2;,I;]
```

```
[0]   TRANSITIONS
[1]   ⎕PW CT FL PN
```

```
[0]   X←TRI M;I
[1]   ⍝ ORDONNE LIGNES, SENS CROISSANT.
[2]   I←(⍴M)[2]
[3]   X←⍳(⍴M)[1]
[4]   DEB:→(0≥I←I-1)/0
[5]   X←X[⍋M[X;I]]
[6]   →DEB
```

```
[0]   T←TS M
[1]   ⍝ M EST UN VECTEUR
[2]   T←(ES[1;;]∧.≤M)/⍳(⍴ES)[2]
```

```
[0]   K←TSAFE
[1]   ⍝ CHECKS TSAFENESS
[2]   →(K←~∨/(2×ES[1;;])∧.≤M)/0
[3]   'CLASS ',(⍕P),' FAILED TSAFE CONDITION'
```

```
[ 0]  B←TS2 M;V;T
[ 1]  ⍝ BAG DES TRANSITIONS SENSIBILISEES
[ 2]  B←⍳0
[ 3]  T←TS M
[ 4]  →(0=⍴T)/0
[ 5]  V←ES[1;T;]
[ 6]  M←(((⍴V)⍴M)×V≠0)+(⌈/M)×V=0
[ 7]  V←⌊/[2]⌊M÷(V=0)+V
[ 8]  B←T
[ 9]  →(∧/V≤1)/0
[10]  V←,⍉(((⌈/V),⍴T)⍴⍉V)×⍉V∘.≥⍳⌈/V
[11]  B←(V≠0)/V
```

```
[ 0]  P UPDATECFCM P;I;V;W
[ 1]  ⍝ CALCUL INCREMENTAL CFCMS
[ 2]  →(P>0)/NEW
[ 3]  P←-P
[ 4]  →(P=P)/SAME
[ 5]  ⍝ P ANCIEN
[ 6]  →(CC[P]=0)/LAB
[ 7]  ⍝ P APPARTIENT A UNE CFCM
[ 8]  I←CC[SK]⍳CC[P]
[ 9]  →GO
[10]  LAB:I←SK⍳P
[11]  GO:→(I>⍴SK)/0
[12]  V←(I-1)↓SK
[13]  W←CC[V]=0
[14]  V←(W/V),(CC∈CC[(~W)/V])/⍳⍴CC
[15]  CC[V]←⌊/V
[16]  →0
[17]  SAME:CC[(⍳⍴CC)⍳P]←P
```

WRITETRANS

% à modifier (cas ∞-enabled)

[12] B ← B × ∧/[2] ES[1; B;] =

B × ∧/[2] (

elems ∧/[2] ES[1; B;] =

```
[5]   LIBOUIDEQNESSIUEST←1↑⍴V
[18]  →0
[19]  NEW:CC←CC,0
```

en cours.

```
[0]   S←UPRED E
[1]   ⍝ E UNION PRED E DANS SG
[2]   S←REDUC E,(∨/[2] SG[1;;]∈E)/⍳(⍴SG)[2]

[0]   S←USUCC E
[1]   ⍝ E UNION SUCC E DANS SG
[2]   S←REDUC E,(,SG[1;E;]≠0)/,SG[1;E;]

[0]   WRITENET
[1]   ⍝ BUILDS AND SAVES DEFINITION OF NET
[2]   ⍝ USEFUL WHEN NET IS BUILT USING MERGE
[3]   'DEFINITION OF ',(⎕FX NETDEF),' SAVED'
[4]   SYSΔNETS←SYSΔNETS EL((⍴NN),1)⍴NN
[5]   SYSΔNETS←SYSΔNETS CONCAT1 NN

[ 0]  ZUBEK
[ 1]  BT1[0,0]: P3, P8 → PT1
[ 2]  ET1[3,3]: PT1 → P1, P9
[ 3]  BT2[0,0]: P6, P7 → PT2
[ 4]  ET2[3,3]: PT2 → P4, P8
[ 5]  BT3[0,0]: P5, P1 → PT3
[ 6]  ET3[1,1]: PT3 → P2, P6
[ 7]  BT4[0,0]: P4 → PT4
[ 8]  ET4[3,3]: PT4 → P5
[ 9]  BT5[0,0]: P9 → PT5
[10]  ET5[4,4]: PT5 → P7
[11]  BT6[0,0]: P2 → PT6
[12]  ET6[2,2]: PT6 → P3
[13]  M0=P1,P5,P7
```

```
[0]   R←L AL L1
[1]   ⍝ AJOULE L1 A MATRICE L
[2]   R←REDUC2 L CONCAT1 L1
```

VLADD

```
[ 0]  TM BE ES;M;V;TN
[ 1]  ⍝ UPDATE TN, ES, IS
[ 2]  TN←TM[1;;]
[ 3]  TN←(∨/[1] TN≠' ')/[2] TN
[ 4]  M←TN CONCAT1 TN
[ 5]  TN←M[(-(⍴TN)[1])↑⍳(⍴M)[1];]
[ 6]  TN←M[REDUC M IOTA M;]
[ 7]  V←(⍳(⍴TN)[1])∈⍳(⍴ES)[2]
[ 8]  ES←V\[2] ES
[ 9]  IS←V\[2] IS
[10]  V←TN IOTA TN
[11]  ES←(2,(⍴TN)[1],(⍴ES)[3])⍴(,((⍴TN)[1],(⍴ES)[3])⍴,ES[1;;]),,((⍴TN)[1],(⍴ES)[3])⍴,ES[2;;]
[12]  ES[;V;]←ES[;V;]+ES
[13]  IS[;V;]←(2,(⍴TM)[2],1)⍴⍬,TM[2 3 ;;],[3](2,(⍴TM)[2],1)⍴' '
[14]  ⍝ CHECK POINT
```

```
[ 0]  M←X CT L;U;V
[ 1]  ⍝ FORMATTE LIGNE EN MATRICE LARGEUR X,
[ 2]  ⍝ L EST NORMALISEE.
[ 3]  X←1⌈X
[ 4]  M←(0,X)⍴''
[ 5]  V←(¯1+(L=' ')/⍳⍴L),⍴L
[ 6]  DEB:→(0=⍴L)/0
[ 7]  U←(V≤X)/V
[ 8]  U←(U$0)/U
[ 9]  →(0=⍴U)/LAB
[10]  M←M,[1] X↑(U←¯1↑U)↑L
[11]  L←(U+1)↓L
[12]  V←V-U+1
```

LMAKETX

```
[ 9]   IS←V\[2] IS
[10]   V←TN IOTA TN
[11]   ES←(2,(ρTN)[1],(ρES)[3])ρ(,((ρTN)[1],(ρES)[3])ρ,ES[1;;]),,((ρTN)[1], … )ρ,ES[2;;]
[12]   ES[;V;]←ES[;V;]+ES
[13]   IS[;V;]←(2,(ρTM)[2],1)ρ⍎,TM[2 3 ;;],[3](2,(ρTM)[2],1)ρ' '
[14]   ⍝ CHECK POINT
```

L MAKE TX

```
[ 0]   M←X CT L;U;V
[ 1]   ⍝ FORMATTE LIGNE EN MATRICE LARGEUR X.
[ 2]   ⍝ L EST NORMALISEE.
[ 3]   X←1⌈X
[ 4]   M←(0,X)ρ''
[ 5]   V←(¯1+(L=' ')/⍳ρL),ρL
[ 6]   DEB:→(0=ρL)/0
[ 7]   U←(V≤X)/V
[ 8]   U←(U≠0)/U
[ 9]   →(0=ρU)/LAB
[10]   M←M,[1] X↑(U←¯1↑U)↑L
[11]   L←(U+1)↓L
[12]   V←V-U+1
[13]   →DEB
[14]   LAB:M←M,[1] X↑L
[15]   L←X↓L
[16]   V←V-X
[17]   →DEB
```

VL DEL

```
[0]   R←L EL L1
[1]   ⍝ SUPPRIME L1 DE MATRICE L
[2]   R←L
[3]   →((ρL1)[2]>(ρL)[2])/0
[4]   L1←((⍳(ρL)[2])∊⍳(ρL1)[2])\[2] L1
[5]   R←(~(⍳(ρL)[1])∊L IOTA L1)/[1] L
```

VL MAKE L

```
[0]   L←LN MAT
[1]   ⍝ MATRICE DE NOMS → LISTE DE NOMS
[2]   L←NR,MAT,[2](ρMAT)[1]ρ' '
```

READMAR

```
[ 0]   MA PV;M;V;K;PN
[ 1]   ⍝ ANALYSE LEXICALE ET STOCKAGE MARQUAGE INITIAL.
[ 2]   PV←(PV≠' ')/PV
[ 3]   PV[(PV=',')/⍳ρPV]←' '
[ 4]   PV←NL PV
[ 5]   V←(∧/[2] PV≠'(')/⍳(ρPV)[1]
[ 6]   M←,PV
[ 7]   M[(M∊'()')/⍳ρM]←' '
[ 8]   PV←(ρPV)ρM
[ 9]   M←((ρPV)[1],2)ρ' '
[10]   M[V;2]←'1'
[11]   PV←PV,[2] M
[12]   M←SP PV
[13]   PN←M[1;;]
[14]   PN←(∨/[1] PN≠' ')/[2] PN
[15]   K←PN CONCAT1 PN
[16]   PN←K[REDUC K IOTA K;]
[17]   PN←K[(-(ρPN)[1])↑⍳(ρK)[1];]
[18]   K←(⍳(ρPN)[1])∊⍳(ρES)[3]
[19]   ES←K\[3] ES
[20]   MO←K\MO
[21]   K←PN IOTA PN
[22]   MO[K]←MO[K]+⍎LN M[2;;]
```

L MAKE VL

```
[ 0]   M←NL L;DML;FML;DMW;FMW;X;N;W1;W2;U
[ 1]   ⍝ FORMATTE LISTE DE MOTS SEPARES PAR UN BLANC EN
[ 2]   ⍝ MATRICE AVEC UN MOT PAR LIGNE CALE A GAUCHE
[ 3]   DML←((' ',L)=' ')/⍳ρ' ',L
[ 4]   FML←(((L,' ')=' ')/⍳ρL,' ')-1
[ 5]   N←ρDML
[ 6]   X←⌈/U←1+FML-DML
[ 7]   DMW←1+X×+¯1+⍳N
[ 8]   FMW←U+DMW
[ 9]   W1←(1+N×X)ρ0
[10]   W1[DMW]←1
[11]   W2←(1+N×X)ρ0
[12]   W2[FMW]←1
[13]   M←(N,X)ρ(¯1↓(+\W1)≠+\W2)\(L≠' ')/L
```

Σ simpl…

~~…~~

Exprimer en listes …

avec |ⁿ de ma…

\+ analyseurs

données par…

```
[5]   LABOLIDEUNESS[EST]←1↑⍴V
```

MAKEL

```
[0]   S←NR L
[1]   ⍝ NORMALISE BLANCS.
[2]   S←(∨\L≠' ')/L
[3]   S←(⌽∨\⌽S≠' ')/S
[4]   L←S≠' '
[5]   S←(L∨1↓1⌽1,L)/S
```

READPLIST

```
[ 0]  M←P PV;V;M
[ 1]  ⍝ ANALYSE LEXICALE LISTE DE PLACES.
[ 2]  PV←,PV
[ 3]  M← 2 0 0 ⍴''
[ 4]  →(0=⍴PV)/0
[ 5]  PV[(PV=',')/⍳⍴PV]←' '
[ 6]  PV←NL PV
[ 7]  V←(∧/[2] PV≠'×')/⍳(⍴PV)[1]
[ 8]  M←,PV
[ 9]  M[(M='×')/⍳⍴M]←' '
[10]  PV←(⍴PV)⍴M
[11]  M←((⍴PV)[1],2)⍴'  '
[12]  M[V;2]←'1'
[13]  PV←PV,[2] M
[14]  M←SP PV
[15]  ⍝ CHECK POINT
```

VL RIGHTJUST

```
[0]   M←RD T
[1]   ⍝ RECALE MATRICE T A DROITE
[2]   M←(⌽-(⍴T)[2]-⌈/[2](T≠' ')×(⍴T)⍴⍳(⍴T)[2])⌽[2] T
```

VL LEFTJUST

```
[0]   M←LG T
[1]   ⍝ RECALE MATRICE T A GAUCHE
[2]   M←(¯1+L/[2]((T=' ')×(⍴T)[2])+(T≠' ')×(⍴T)⍴⍳(⍴T)[2])⌽[2] T
```

[redacted]

MW EXPLODE

```
[0]   T←SP M
[1]   ⍝ X MOTS PAR LIGNE DANS M;
[2]   ⍝ T: TABLEAU A X PLANS AVEC MOTS SEPARES.
[3]   T←NL NR,M,[2]((⍴M)[1],1)⍴' '
[4]   T← 2 1 3 ⍉((⍴M)[1],((⍴T)[1]÷(⍴M)[1]),(⍴T)[2])⍴,T
```

LSUBST

```
[ 0]  V←AB SU L;A;B;X;Y
[ 1]  ⍝ SUBSTITUTION, AB FORMAT '(A⍪B)'.
[ 2]  V←AB⍪'|'
[ 3]  A←(V-1)↑AB
[ 4]  B←(V-⍴AB)↑AB
[ 5]  Y←(∧/[1](¯1+⍳⍴B)∘.⌽B∘.=L)/⍳⍴L
[ 6]  V←L
[ 7]  →(0=⍴Y)/0
[ 8]  V←⍳0
[ 9]  DEB:→(0=⍴Y)/END
[10]  X←¯1↑Y
[11]  Y←¯1↓Y
[12]  V←A,((⍴B)↓(¯1+X-⍴L)↑L),V
[13]  L←(X-⍴L)↓L
[14]  →DEB
[15]  END:V←((X-1)↑L),V
```

```
[ 0]  M←T TV;V;M
```

```
[ 5]   X←(∧/[1](¯1+⍳ρB)∘.≤B∘.=L)/⍳ρL
[ 6]   V←L
[ 7]   →(0=ρX)/0
[ 8]   V←⍳0
[ 9]   DEB:→(0=ρX)/END
[10]   X←¯1↓X
[11]   J←¯1↓X
[12]   V←A,((ρB)↑(¯1+X-ρL)↑L),V
[13]   L←(X-ρL)↓L
[14]   →DEB
[15]   END:V←((X-1)↑L),V
```

READTLIST

```
[ 0]   M←T TV;V;N
[ 1]   ⍝ ANALYSE LEXICALE LISTE DE TRANSITIONS.
[ 2]   TV←,TV
[ 3]   V←~(+\TV='[')-+\TV=']'
[ 4]   V←V∧TV=','
[ 5]   TV[V/⍳ρTV]←' '
[ 6]   TV←'¯1|ω' SU TV
[ 7]   TV←NL TV
[ 8]   V←(∧/[2] TV≠'[')/⍳(ρTV)[1]
[ 9]   M←,TV
[10]   M[(M∈'[,]')/⍳ρM]←' '
[11]   TV←(ρTV)ρM
[12]   M←((ρTV)[1],5)ρ' '
[13]   M[V;]←((ρV),5)ρ' 0 ¯1'
[14]   TV←TV,[2] M
[15]   M←SP TV
[16]   ⍝ CHECK POINT
```

```
[ 0]   ES←N1 TO N2;W1;W2;M;K
[ 1]   ⍝ UPDATE PN, PRODUIT LIGNE DE ES
[ 2]   W1←1↓£,'0 ',LN N1[2;;]
[ 3]   N1←N1[1;;]
[ 4]   N1←(∨/[1] N1≠' ')/[2] N1
[ 5]   W2←1↓£,'0 ',LN N2[2;;]
[ 6]   N2←N2[1;;]
[ 7]   N2←(∨/[1] N2≠' ')/[2] N2
[ 8]   M←PN CONCAT1 N1 CONCAT1 N2
[ 9]   N2←M[⍳(-(ρN2)[1])↑⍳(ρM)[1];]
[10]   N1←M[⍳(ρN1)[1]↑⍳(-(ρN1)[1]+(ρN2)[1])↑⍳(ρM)[1];]
[11]   PN←M[REDUC M IOTA M;]
[12]   ES←(K←(⍳(ρPN)[1])∈⍳(ρES)[3])\[3] ES
[13]   MO←K\MO
[14]   ES←(2¯1 ,(ρPN)[1])ρ0
[15]   ES[1;;PN IOTA N1]←W1
[16]   ES[2;;PN IOTA N2]←W2
[17]   ⍝ CHECK POINT
```