

Zeitschrift: Bulletin de la Société Vaudoise des Sciences Naturelles
Herausgeber: Société Vaudoise des Sciences Naturelles
Band: 76 (1982-1983)
Heft: 361

Artikel: Recherche automatique des associations unitaires en biochronologie
Autor: Guex, Jean / Davaud, Eric
DOI: <https://doi.org/10.5169/seals-278145>

Nutzungsbedingungen

Die ETH-Bibliothek ist die Anbieterin der digitalisierten Zeitschriften auf E-Periodica. Sie besitzt keine Urheberrechte an den Zeitschriften und ist nicht verantwortlich für deren Inhalte. Die Rechte liegen in der Regel bei den Herausgebern beziehungsweise den externen Rechteinhabern. Das Veröffentlichen von Bildern in Print- und Online-Publikationen sowie auf Social Media-Kanälen oder Webseiten ist nur mit vorheriger Genehmigung der Rechteinhaber erlaubt. [Mehr erfahren](#)

Conditions d'utilisation

L'ETH Library est le fournisseur des revues numérisées. Elle ne détient aucun droit d'auteur sur les revues et n'est pas responsable de leur contenu. En règle générale, les droits sont détenus par les éditeurs ou les détenteurs de droits externes. La reproduction d'images dans des publications imprimées ou en ligne ainsi que sur des canaux de médias sociaux ou des sites web n'est autorisée qu'avec l'accord préalable des détenteurs des droits. [En savoir plus](#)

Terms of use

The ETH Library is the provider of the digitised journals. It does not own any copyrights to the journals and is not responsible for their content. The rights usually lie with the publishers or the external rights holders. Publishing images in print and online publications, as well as on social media channels or websites, is only permitted with the prior consent of the rights holders. [Find out more](#)

Download PDF: 14.04.2026

ETH-Bibliothek Zürich, E-Periodica, <https://www.e-periodica.ch>

BULLETIN N° 261 des Laboratoires de Géologie, Minéralogie, Géophysique
et du Musée géologique de l'Université de Lausanne

Recherche automatique des associations unitaires en biochronologie¹

PAR

JEAN GUEX² et ERIC DAVAUD³

Résumé. Le programme FORTRAN ASCII présenté dans cette note permet d'établir des *range charts* à partir d'informations stratigraphiques provenant d'un nombre illimité de profils mais n'impliquant pas plus de 110 espèces. Les démarches algorithmiques sur lesquelles il est basé sont une version améliorée de celles qui ont été publiées par GUEX (1980, 1981).

INTRODUCTION

Un modèle analytique qui permet de résoudre efficacement (mais lentement) le problème général posé par l'utilisation des espèces fossiles comme outil de datation a été élaboré dans une série d'articles publiés ici même (GUEX, 1977-1981). La conclusion naturelle de ces recherches est une version informatisée des démarches exposées dans les travaux susmentionnés: cet aboutissement est présenté aujourd'hui.

Seuls les éléments essentiels à la compréhension du programme seront précisés, le lecteur étant renvoyé aux publications citées plus haut pour les détails techniques des opérations qui n'ont pas été modifiées ou améliorées ici.

Les modifications de procédure qui ont été effectuées sont de deux sortes:

1) *Algorithme de détection des sous-graphes Z_4 , S_3 , S_4 et ordre des opérations*

GUEX (1980) propose de détruire les cycles Z_4 avant de calculer les cliques maximales, les sous-graphes S_3 et S_4 résiduels étant détectés ultérieure-

¹Travail effectué dans le cadre des projets 2.685.082 et 2.667.080 du Fonds national de la recherche scientifique.

²Institut de Géologie, Palais de Rumine, 1005 Lausanne.

³Département de Géologie, Maraîchers 13, 1211 Genève 4.

ment. Pour des raisons d'optimisation, l'ordre de ces opérations a été inversé: on commence par détecter les circuits semi-orientés S_3 et S_4 avant de chercher les cycles Z_4 et avant de déterminer les cliques. Il découle de cela que tous les Z_4 de type «2» (*loc. cit.* p. 118, fig. 1-e) sont détectés et détruits en même temps que les S_4 . L'algorithme utilisé ici pour la détection de ces sous-graphes peut être résumé ainsi: pour tout couple de sommets du graphe G reliés par une arête (i, j) on cherche l'intersection entre l'ensemble des prédécesseurs et celui des successeurs du couple :

$$C = \Gamma_{(ij)}^- \cap \Gamma_{(ij)}^+ \quad \text{où} \quad \Gamma_{(ij)}^\pm = \Gamma_i^\pm \cup \Gamma_j^\pm$$

Si $C = \emptyset$, l'arête (i, j) se trouve impliquée dans m S_3 , où m est la cardinalité de l'ensemble C .

Dans un deuxième temps (et quelle que soit la valeur de m), on cherche le complémentaire de C dans l'ensemble des prédécesseurs puis dans celui des successeurs du couple initial (i, j) :

$$A = \left(C \right)^{\Gamma_{(ij)}^-} ; \quad B = \left(C \right)^{\Gamma_{(ij)}^+}$$

On vérifiera aisément que le couple (i, j) est impliqué dans un S_4 ou dans un Z_4 de type «2» s'il existe au moins une arête entre les éléments de A et ceux de B .

2) *Préservation optimale des arcs*

Un remplacement systématique des arcs impliqués dans les sous-graphes Z_4 , S_3 et S_4 (= configurations interdites) par des arêtes virtuelles peut induire une perte importante dans les informations chronologiques potentiellement contenues dans le graphe de départ. Dans le programme ci-dessous, la substitution de ces arcs par des arêtes se fait de manière *différée* et *sélective*. Elle intervient seulement lorsque l'inventaire de ces configurations interdites est établi et que les intrications entre les arcs impliqués sont mémorisées dans une matrice $A = (a_{kl})$ où

$$a_{kl} = \begin{cases} 1 & \text{si l'arc } k \text{ est conjointement impliqué avec l'arc } l \text{ dans une} \\ & \text{configuration interdite} \\ 0 & \text{sinon} \end{cases}$$

Deux critères sont utilisés pour détruire un minimum d'arcs (i. e. pour conserver un maximum d'informations sur les relations de superposition entre les espèces) tout en éliminant un nombre optimal de configurations interdites:

a) Les arcs les plus fréquemment impliqués dans les sous-graphes susmentionnés sont remplacés progressivement par des arêtes, la matrice A étant réajustée à chaque itération.

b) Lorsque deux arcs présentent la même fréquence d'implication, on détruit préférentiellement celui qui est le moins reproductible (i. e.

celui qui correspond à la relation stratigraphique la moins fréquemment observée).

Au terme de cette procédure, toutes les configurations détectées au cours de la première itération sont détruites, mais les arêtes nouvelles résultant de la destruction des arcs peuvent engendrer de nouvelles configurations interdites. Il est donc nécessaire de répéter l'ensemble des opérations ci-dessus jusqu'à ce que cette manipulation cesse d'engendrer des configurations interdites de $n^{\text{ième}}$ génération.

DESCRIPTION DU PROGRAMME (Tables I-a à I-i)

Le programme listé à la fin de ce travail permet d'établir des *range charts* à partir d'informations stratigraphiques provenant d'un nombre illimité de profils mais n'impliquant pas plus de 110 espèces. Cette limite peut être élargie en modifiant les dimensions des matrices et vecteurs déclarés en début de programme.

Les temps d'exécution ne sont pas influencés par le nombre de profils. Par contre, ils augmentent en fonction du nombre d'espèces et surtout du nombre de configurations interdites S_3 , S_4 et Z_4 .

A titre indicatif, la résolution du problème «Alvéolines» (fig. 1) ne requiert que quelques secondes alors que celle du problème «Nannoplankton» (fig. 3) (où le graphe G contient plus de 2600 configurations interdites distinctes) exige environ 4 minutes de calcul sur UNIVAC 1100/60...

Le mode de soumission des données est illustré par le jeu d'essai (fig. 1) et brièvement commenté ici. Les espèces doivent préalablement être numérotées consécutivement de 1 à n ($n \leq 110$). On indique d'abord le titre du problème (76 caractères au maximum), le nombre total de profils et le nombre total d'espèces. Les données proprement dites sont fournies profil par profil dans un ordre quelconque. Pour chaque profil on indique le nombre d'espèces localement observées, puis l'extension stratigraphique locale de celles-ci en indiquant la cote (numérotée à partir du bas) de leur première apparition et celle de leur disparition.

Toutes les informations numériques sont données en format libre mais doivent être exprimées en nombres entiers (INTEGER).

L'impression de tous les tableaux et messages intermédiaires peut être obtenue en laissant vide la première colonne du titre.

Ce programme, écrit en FORTAN ASCII, occupe environ 50 k-mots mémoire et utilise deux fichiers temporaires (unité 10 et 12). Il fait appel à une pseudo-fonction BITS spécifique au software UNIVAC: l'utilisateur se référera à la description qui en est donnée au début de la sous-routine STR pour en trouver l'équivalent sur un autre type d'ordinateur (p. ex. sous-routine FLD sur HONEYWELL BULL et sous-routine SHIFT sur CDC).

JEU D'ESSAI: DISTRIBUTION BIOCHRONOLOGIQUE DES ALVÉOLINES ILERDIENNES
DE YOUGOSLAVIE (DROBNE 1977)

Le problème analysé dans le jeu d'essai (fig. 1) a récemment été étudié par GUEX (1981). Le lecteur est donc renvoyé aux données de base (tirées de DROBNE, 1977) compilées dans cette note: il y trouvera la correspondance entre les numéros assignés aux profils stratigraphiques (1 à 11) et leurs noms respectifs, ainsi que la correspondance entre les numéros des espèces (1 à 15) et leurs noms (*loc. cit.* p. 181).

Commentaires

L'output 1 (fig. 1-a et b) montre deux *range charts* distinctes. La première correspond au résultat brut obtenu à l'aide du programme. Cette *range chart* est ensuite épurée pour mettre en évidence d'éventuelles discontinuités dans la distribution biochronologique des espèces (discussion détaillée in GUEX, 1980, p. 124, et 1981, p. 185). L'examen de la *range chart* 2 (fig. 1-b) qui résulte de cette épuration fait apparaître immédiatement une discontinuité dans la distribution de l'espèce 3: celle-ci est absente dans tous les niveaux où l'association unitaire N° 3 peut être identifiée. Or, la table de correspondance entre les niveaux fossilifères des différents profils et les associations unitaires qui y sont identifiables (fig. 1-c) montre que cette association N° 3 est identifiable de manière reproductible (on la trouve dans les profils 2, 3, 6, 7 et 8). On en déduit que l'espèce 3 peut avoir subi un remaniement: en se référant aux données stratigraphiques de base (GUEX, 1981, p. 181, fig. 1), on trouve sans peine que cette espèce occupe une position anormale dans l'échantillon 5 (= niv. 4) du profil 1. Si l'on traite le problème après avoir retiré cet échantillon des données de base, on obtient alors l'output 2 (fig. 2) qui est une solution conforme à celle de DROBNE (1977) et de GUEX (1981, p. 187, fig. 6-h).

DISTRIBUTION BIOCHRONOLOGIQUE
DU NANNOPLANCTON PALÉOGÈNE DE CALIFORNIE

L'exemple donné ci-dessous concerne la zonation du Paléogène de Californie découlant des recherches de BRAMLETTE & SULLIVAN (1961) et SULLIVAN (1965) sur la distribution stratigraphique de plus de 100 espèces de nannoplancton calcaire. Ce problème a été discuté avec un certain détail dans la publication de DAVAUD & GUEX (1978) où le lecteur trouvera une compilation des données biostratigraphiques de base (*loc. cit.* Planche hors-texte, dans laquelle les profils A à J correspondent aux profils 1 à 10 de la présente note), ainsi que la correspondance entre les numéros des espèces (1 à 104) et leurs noms (*loc. cit.* p. 595).

INPUT

TITRE (col. 2 - 77) — JEU D'ESSAI: ALVEOLINES (DROBNE)
 11 PROFILS TOTALISANT 15 — 11 15
 ESPECES
 LE PREMIER PROFIL CONTIENT 7 ESPECES :
 L'ESPECE 2 NE S'OBSERVE QUE DANS LE NIVEAU 1 (2 1 1), L'ESPECE 3 DANS LE NIVEAU 4, ETC...

```

7
2 1 1 3 4 4 4 3 3 6 2 2 7 3 3 8 4 4 12 3 3
10
1 3 3 3 1 1 5 1 1 6 2 3 7 1 3 9 3 4 11 3 3 12 3 3 13 2 2 14 4 4
11
2 1 1 3 2 2 6 2 3 7 2 3 9 6 7 10 7 7 11 5 5 12 4 4 13 3 3 14 6 7 15 4 6
10
1 1 2 4 3 3 7 1 1 8 2 2 9 1 4 10 3 4 11 1 2 12 2 2 14 3 4 15 4 4
2
4 2 2 5 1 1
7
1 2 2 4 3 3 7 1 2 9 3 3 10 4 4 13 1 1 14 4 4
6
2 1 1 3 3 3 5 2 2 7 3 4 13 4 4 15 4 4
8
2 1 1 5 1 1 6 3 3 7 2 2 8 5 5 10 6 6 13 4 4 14 5 6
4
6 2 2 10 1 2 12 2 2 15 1 1
6
1 1 1 6 1 1 8 3 3 10 2 2 11 1 1 12 1 1
3
9 1 1 10 1 1 14 1 1
    
```

OUTPUT 1

RANGE CHART 1															RANGE CHART 2																		
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15				
1:	0	0	0	1	0	0	0	1	1	1	0	0	0	1	1	1:	0	0	0	1	0	0	0	1	1	1	0	0	0	1	1		
2:	1	0	1	1	0	1	1	1	1	1	1	0	0	0	1	2:	1	0	1	1	0	1	1	1	1	1	1	1	1	1	0	0	1
3:	0	0	1	0	0	1	1	0	0	0	0	0	1	0	1	3:	0	0	0	0	0	1	1	0	0	0	0	0	0	0	1	0	1
4:	0	1	1	0	1	0	1	0	0	0	0	0	0	0	0	4:	0	1	1	0	1	0	1	0	0	0	0	0	0	0	0	0	0

a

b

PROFIL : 1
 NIVEAU: 1 3 4
 ASS. UN. 4 2 2

PROFIL : 5
 NIVEAU: 1
 ASS. UN. 4

PROFIL : 8
 NIVEAU: 1 4 5 6
 ASS. UN. 4 3 1 1

PROFIL : 2
 NIVEAU: 1 2 3 4
 ASS. UN. 4 3 2 1

PROFIL : 6
 NIVEAU: 1 2 4
 ASS. UN. 3 2 1

PROFIL : 9
 NIVEAU: 2
 ASS. UN. 2

PROFIL : 3
 NIVEAU: 1 3 4 5 6 7
 ASS. UN. 4 3 2 2 1 1

PROFIL : 7
 NIVEAU: 1 2 4
 ASS. UN. 4 4 3

PROFIL : 10
 NIVEAU: 1
 ASS. UN. 2

PROFIL : 4
 NIVEAU: 1 2 3 4
 ASS. UN. 2 2 1 1

PROFIL : 11
 NIVEAU: 1
 ASS. UN. 1

c

Fig. 1. — Jeu d'essai (détails: cf. texte)

a) Range chart brute montrant la distribution des espèces 1 à 15 dans les associations unitaires 4 à 1.

b) Range chart épurée dans le but de mettre en évidence des discontinuités dans la distribution biochronologique des espèces.

c) Table de correspondance «associations unitaires-niveaux».

de SULLIVAN (caractérisée par les espèces 30, 36, 49, 72 et 82) correspond à l'A.U.5 et ses zones 3 et 4 recouvrent respectivement les A.U. 4+3 et 2+1.

Le traitement automatique de ce problème confirme donc une fois de plus la qualité de l'analyse biochronologique empirique de SULLIVAN et il corrobore implicitement les corrélations que MALLORY (1959) a établies sur la base de la distribution stratigraphique des foraminifères benthiques (détails in GUEX, 1979, p. 191) dans les mêmes localités.

2) «Pseudo-discontinuités»

Certaines espèces (p. ex. 97, 102 et 104) montrent une distribution apparemment discontinue dans la *range chart* illustrée ci-dessus (fig. 3-a). Un examen sommaire de la matrice A.U. – Profils (fig. 3-c) permet toutefois de constater que les associations unitaires dans lesquelles ces espèces sont absentes (p. ex. A.U. 8 et 7) ne sont pas reproductibles. Ces discontinuités ne peuvent donc pas être interprétées en termes de remaniements: elles sont clairement dues à des facteurs documentaires.

BIBLIOGRAPHIE

- BRAMLETTE, M.N. & SULLIVAN, F. R. 1961. – Coccolithophorids and related nannoplankton of the early Tertiary in California. *Micropaleontology* 7/2.
- DAVAUD, E. & GUEX, J. 1978. – Traitement analytique «manuel» et algorithmique de problèmes complexes de corrélations biochronologiques. *Ecl. Geol. Helv.* 71/3.
- DROBNE, K. 1977. – Alvéolines paléogènes de la Slovénie et de l'Istrie. *Mém. suisses pal.* 99.
- GUEX, J. 1977. – Une nouvelle méthode d'analyse biochronologique. *Bull. Soc. vaud. Sc. nat.* 73/3 (N° 351) et *Bull. Géol. Univ. Lausanne*, 224.
- 1979. – Terminologie et méthodes de la biostratigraphie moderne. *Bull. soc. vaud. Sc. nat.* 74/3 (N° 355) et *Bull. Géol. Univ. Lausanne* 234.
- 1980. – Calcul, caractérisation et identification des associations unitaires en biochronologie. *Bull. Soc. vaud. Sc. nat.* 75/2 (N° 358) et *Bull. Géol. Univ. Lausanne* 249.
- 1981. – Associations virtuelles et discontinuités dans la distribution des espèces fossiles. *Bull. Soc. vaud. Sc. nat.* 75/3 (N° 359) et *Bull. Géol. Univ. Lausanne* 255.
- MALLORY, V. S. 1959. – Lower Tertiary biostratigraphy of the California Coast Ranges. *Amer. Ass. Petrol. Geol. Tulsa*.
- SULLIVAN, A. B. 1965. – Lower Tertiary Nannoplankton from the California Coast Ranges. *Univ. Calif. Publ. geol. Sci.* 53.

Manuscrit reçu le 10 mars 1982.

TABLE Ia

```

IMPLICIT LOGICAL (A-H), LOGICAL (R-T)
DIMENSION NE(110), LI(110), LS(110), B(110), F(110), Q(110),
*H(110), AI(72, 110), NDA(110), NOB(200), E(110), IFR(8000), AR(110, 110)
EXTERNAL AT
COMMON/MATRIX/T(8000)
COMMON/TTR/TITL(19)
COMMON/TEST/IARC(3000), IART(3000), NARC, NART
COMMON/MATRAS/AS(110, 110)
CHARACTER PPR*1
EQUIVALENCE (IFR, AS), (AI, T)
COMMON/DAT/MMAX, NBIT
DATA NBIT/36/MMAX/750/ @ MOTS DE 36 BITS , DIMENSION MAXIMALE MATRICE AT
C
READ 100, PPR, TITL
100 FORMAT(A1, 19A4)
APR=. TRUE.
IF (PPR, NE, 1H ) APR=. FALSE.
READ 101, NC, M
101 FORMAT()
NM=(M*M-M)/2+M
DO 136 I=1, NM
136 IFR(I)=0
CALL INIT(T, .FALSE., 100)
DO 172 I=1, M
172 NDA(I)=0
DO 1 I=1, NC @ LECTURE DES DONNEES SECTION PAR SECTION
READ 101, MI
READ 101, (NE(K), LI(K), LS(K), K=1, MI)
WRITE(10) MI, (NE(K), LI(K), LS(K), K=1, MI)
DO 2 J=1, MI
NDA(NE(J))=NDA(NE(J))+1
DO 2 L=1, J
K=MIN(NE(J), NE(L))
IA=(K-1)*M-((K-1)**2+K-1)/2+MAX(NE(J), NE(L))
IF (LS(J)-LI(L).LT.0) GO TO 133
AR(NE(J), NE(L))=. TRUE. @CONSTRUCTION DE LA MATRICE ASSOCIEE AU GRAPHE Q
IFR(IA)=IFR(IA)+1
133 IF (LS(L)-LI(J).LT.0) GO TO 134
AR(NE(L), NE(J))=. TRUE.
IFR(IA)=IFR(IA)+1
134 IF ((LS(J)-LI(L))*(LS(L)-LI(J)).LT.0) GO TO 2
AR(NE(J), NE(L))=. TRUE.
AR(NE(L), NE(J))=. TRUE.
2 CONTINUE
1 CONTINUE
WRITE(12)(IFR(I), I=1, NM) @ FREQUENCE DES ARCS OBSERVES. MEMORISATION
REWIND 12
DO 137 I=1, NM
137 IFR(I)=0
ICL=0
IF (APR) CALL PRTN(110, 110, AR, M, M, 'MATRICE DES RELATIONS ENTRE ESP
*ECES ')
C..... DETECTION DES S3, S4 ET CERTAINS Z4 DANS LE GRAPHE Q ET CONSTRUCTION
C DU GRAPHE DES ARCS IMPLIQUES DANS CES CONFIGURATIONS
47 NARC=0
ICL=ICL+1 @NOMBRE D'ITERATIONS
DO 177 I=1, M
177 LI(I)=0
NART=0
DO 58 I=1, M
DO 58 J=1, M
58 AS(I, J)=. FALSE.
NCY=0
DO 40 I=1, M
CALL NEIB(AR, I, G, M)
CALL GAMMA(AR, I, M, B, F) @SUCESSEURS ET PREDECESSEURS DE I
DO 40 J=1, M
IF (I.EQ. J. OR. .NOT. G(J)) GO TO 40
CALL CYCLE(I, J, AR, M, NCY, B, F, LI) @L ARETE I-J EST ELLE IMPLIQUEE DANS CYCLE
40 CONTINUE
IF (NCY.EQ.0) GO TO 42 @ PAS DE S3 NI DE S4 DETECTE
IF (APR) PRINT 456, ICL, NCY
456 FORMAT(//, 1X, 'ITERATION : ', I2, 1X, I5, ' S3, S4 ONT ETE DETECTES')
IF (APR) PRINT 400, (IART(I), I=1, NART)
400 FORMAT(///, 1X, 'STATISTIQUE DES ARETES IMPLIQUEES DANS LES S3, S4, Z4'
*, //, (1X, B(1H(, I3, 1H-, I3, 2H):, I4, 2X)))
IF (APR) PRINT 500
IF (APR) PRINT 401, (IARC(I), I=1, NARC)
401 FORMAT(///, 1X, 'STATISTIQUE DES ARCS IMPLIQUES DANS LES S3, S4, Z4'
*, //, (1X, B(1H(, I3, 1H-, I3, 2H):, I4, 2X)))
IF (APR) PRINT 500
500 FORMAT(///)
IK=NARC/3
C..... ELIMINATION SELECTIVE DE CERTAINS ARCS PAR ADJONCTION D'ARETES
C ET REAJUSTEMENT DU GRAPHE DES ARCS IMPLIQUES
READ(12)(IFR(I), I=1, NM)
DO 142 K=3, NARC, 3
IN=K/3
KG=MIN(IARC(K-2), IARC(K-1))
IA=(KG-1)*M-((KG-1)**2+KG-1)/2+MAX(IARC(K-2), IARC(K-1))
142 IART(IN)=IFR(IA)
REWIND 12
DO 138 I=1, NM
138 IFR(I)=0

```

TABLE Ib

```

54 IM=0
DO 50 I=3, NARC, 3
IF(IARC(I).GT.IM)GO TO 550
IF(IARC(I).LT.IM)GO TO 50
IF(IART(INT(I/3)).GT.IART(IN))GO TO 50 @MAINTIEN DES ARCS LES PLUS
REPRODUCTIBLES
C
550 IM=IARC(I)
IN=I/3
50 CONTINUE
IF(IM.EQ.0)GO TO 42 @ CONFIGURATIONS PROBLEMATIQUES DETRUITES
143 IARC(IN*3)=0
C..... LES NOUVELLES ARETES ENGENDRENT-ELLES DES CONFIGURATIONS PROBLEMATIQUES
I=IARC(IN*3-2)
J=IARC(IN*3-1)
IF(APR)PRINT 402, I, J, IART(IN)
402 FORMAT(1X, 2I5, ' CET ARC EST REMPLACE PAR UNE ARETE. FREQ. D OBSER
*VATION: ', I5)
AR(I, J)=.TRUE. @REMPLACEMENT DE L'ARC I-J PAR UNE ARETE
AR(J, I)=.TRUE.
C..... REAJUSTEMENT DU GRAPHE DES ARCS APRES ELIMINATION DE L'ARC I-J
DO 51 L=1, IK
IF(.NOT.AT(IN, L))GO TO 51
IARC(L*3)=IARC(L*3)-1
CALL STR(IN, L, .FALSE., T) @REINITIALISATION DE LA MATRICE ASSOCIEE
AU GRAPHE DES ARCS IMPLIQUES
C
51 CONTINUE
GO TO 54
42 CONTINUE
C..... DESTRUCTION DE CERTAINS Z4 (SANS ARCS)
23 AW=.FALSE.
MO=M-1
DO 20 I=1, MO
IS=I+1
DO 20 J=IS, M
AZ=.FALSE.
IF(AR(I, J).OR.AR(J, I))GO TO 20 @RECHERCHE SOMMETS SANS ARETE
CALL NEIB(AR, I, G, M) @ VOISIN DE I
CALL NEIB(AR, J, F, M) @ VOISIN DE J
CALL INTR(G, F, B, M) @ VOISINS COMMUNS A I ET J
IF(ICARD(B, M).LE.1)GO TO 20
CALL INDEX(B, 1, M, LS, KA)
KL=KA-1
DO 21 K=1, KL
KO=K+1
DO 22 L=KO, KA
IF(AR(LS(K), LS(L)).AND.AR(LS(L), LS(K)))GO TO 22
AZ=.TRUE.
NCY=NCY+1
IF(AR(LS(K), LS(L)).OR.AR(LS(L), LS(K)))GO TO 22 @MAINTIEN DE L ORIENTATION
AR(LS(K), LS(L))=.TRUE.
AR(LS(L), LS(K))=.TRUE. @ ADJONCTION D ARETE ENTRE VOISINS COMMUNS
IF(APR)PRINT 453, LS(K), LS(L)
453 FORMAT(1X, 'ADJONCTION D ARETE ENTRE : ', 2I6)
22 CONTINUE
21 CONTINUE
IF(.NOT.AZ)GO TO 20 @LE COUPLE I-J NEST PAS IMPLIQUE DANS UN Z4
AW=.TRUE.
AR(I, J)=.TRUE.
AR(J, I)=.TRUE. @ADJONCTION D ARETE ENTRE I ET J
IF(APR)PRINT 453, I, J
20 CONTINUE
IF(AW)GO TO 23
IF(NCY.GT.0)GO TO 47 @UNE NOUVELLE ITERATION EST NECESSAIRE
C..... RECHERCHE DES ASSOCIATIONS UNITAIRES
LT=1
AS(1, 1)=.TRUE.
N=1
11 CONTINUE
N=N+1
IF(N.GT.M)GO TO 10
L=LT
CALL NEIB(AR, N, F, N-1) @VOISIN DE N DANS G(N-1)
IF(ICARD(F, N-1).GT.0)GO TO 30
LT=LT+1 @ N EST UNE NOUVELLE ASSOCIATION UNITAIRE (ASS. UN.)
AS(LT, N)=.TRUE.
GO TO 11
30 DO 4 I=1, L
NO=N-1
DO 5 J=1, NO
5 H(J)=AS(I, J) @ H: IEME ENSEMBLE COMPATIBLE DE G(N-1)
CALL INTR(F, H, E, N-1) @ E: INTERSECTION DE L'ENSEMBLE DES VOISINS DE N ET
DE LA I EME ASS. UN. DU GRAPHE G AU RANG N-1
C
IF(ICARD(E, N-1).EQ.0)GO TO 4
IF(.NOT.EQUAL(E, H, N-1))GO TO 4
AS(I, N)=.TRUE. @ INTERSECTION DE N ET DU I EME ENS. COMP EST UNE NOUVELLE
ASSOCIATION UNITAIRE
C
IF(EQUAL(E, F, N-1))GO TO 78
GO TO 4
6 E(N)=.TRUE. @ E=ENSEMBLE(E) REUNION N : NOUVELLE ASS. UN.
LT=LT+1
DO 9 J=1, N
AS(LT, J)=E(J) @E EST PROVISoireMENT UNE NOUVELLE ASS. UN.
9 CONTINUE
4 CONTINUE

```

TABLE IC

```

78 LTR=0
DO 70 I=1,LT
L=I+1
DO 73 K=1,N
73 H(K)=AS(I,K)
IF(I.EQ.LT)GO TO 77
DO 71 J=L,LT
DO 74 K=1,N
74 G(K)=AS(J,K)
CALL REUN(H,G,F,N) @ REUNION DES ENSEMBLES I ET J
IF(EQUAL(F,G,N))GO TO 70 @L'ENS. I EST INCLU DANS L'ENS. J
IF(.NOT.EQUAL(F,H,N))GO TO 71
DO 75 K=1,N
AS(J,K)=H(K) @ L'ENS. I REMPLACE L'ENS. J
75 CONTINUE
GO TO 70
71 CONTINUE
77 LTR=LTR+1
DO 76 K=1,N
AS(LTR,K)=H(K) @KL(I) EST UNE ASSOCIATION UNITAIRE AU RANG N
76 CONTINUE
70 CONTINUE
LTR=LTR
GO TO 11
C.....ORDONNANCE DES ASSOCIATIONS UNITAIRES ET CONSTRUCTION DU GRAPHE G.K
10 LD=LT-1
CALL PRTN(110,110,AS,LT,M,'ASSOCIATIONS UNITAIRES
*)
C..... IDENTIFICATION DES ASSOCIATIONS UNITAIRES DANS LES SECTIONS
C..... ET CONSTRUCTION DE G.K
CALL IDENT(AS,AR,NC,LT,M,NE,LI,LS,NOA,NOB,B,F,G,AI,.FALSE.)
IF(APR)CALL PRTN(110,110,AR,LT,LT,'MATRICE ASSOCIEE AU GRAPHE G.K
*
')
C..... DETECTION DES CIRCUITS C(N) DANS G.K
CALL TRSF(AR,LT,G,E,F,H)
C..... RECHERCHE DES SUCESSEURS IMMEDIATS ET CORRECTION DE G.K
DO 80 I=1,LT
CALL GAMMA(AR,I,LT,G,H)
DO 82 J=1,LT
82 E(J)=.FALSE.
IF(ICARD(H,LT).EQ.0)GO TO 80
DO 81 J=1,LT
IF(.NOT.H(J))GO TO 81
CALL GAMMA(AR,J,LT,G,F)@SUCESSEUR(S) DES SUCESSEURS DE LA I EME ASS.UN.
CALL REUN(F,E,G,LT)
DO 84 K=1,LT
84 E(K)=G(K)
81 CONTINUE
CALL INTR(E,H,G,LT) @SUCESSEURS DE RANG 2
CALL CDMPL(H,G,E,LT) @ [E] = SUCESSEUR(S) IMMEDIAT(S)
DO 83 J=1,LT
IF(.NOT.E(J))AR(J,I)=.FALSE. @ELIMINATION SUCESSEURS NON IMM.
83 CONTINUE
80 NE(I)=ICARD(E,LT)
IF(APR)CALL PRTN(110,110,AR,LT,LT,' GRAPHE GK SUCESSEURS IMMEDIAT
*S
')
C..... RECHERCHE DE L'ENSEMBLE DES ASSOCIATION UNITAIRES SANS SUCESSEURS
DO 86 I=1,LT
DO 86 J=1,LT
CALL STR1(I,J,.FALSE.,T)
86 CONTINUE
DO 85 J=1,LT
CALL GAMMA(AR,J,LT,G,F)
IF(ICARD(F,LT).EQ.0)CALL STR1(I,J,.TRUE.,T)
85 CONTINUE
C..... RECHERCHE DE LA PLUS LONGUE CHAINE DE G.K : [L]
I=1
KMAX=0
K=0
88 J=0
87 J=J+1
IF(J.LE.LT)GO TO 89
I=I-1
IF(I.EQ.0)GO TO 98
CALL STR1(I,NE(K),.FALSE.,T)
K=K-1
GO TO 88
89 IF(.NOT.AT1(I,J))GO TO 87
K=K+1
NE(K)=J
CALL GAMMA(AR,J,LT,G,F)
IF(ICARD(G,LT).GT.0)GO TO 96
CALL STR1(I,J,.FALSE.,T) @FIN D'UNE CHAINE
IF(APR)PRINT 407,(NE(L),L=1,K)
407 FORMAT(1X,'CHAINE DETECTEE :',(20I3,/,20X))
IF(K.LT.KMAX)GO TO 28
DO 29 IK=1,K
29 LI(IK)=NE(IK) @MEMORISATION DE LA PLUS LONGUE CHAINE DE G.K
KMAX=K
28 K=K-1
GO TO 87
96 I=I+1 @POURSUITE DE LA CHAINE
DO 97 L=1,LT
IF(G(L))CALL STR1(I,L,.TRUE.,T) @ MEMORISATION DES PREDECESSEURS
97 CONTINUE
GO TO 88

```

```

TRANSFORMATION DE G K
98 IF(KMAX.LT.LT)GO TO 158 @IL N'Y A QU'UNE SEULE CHAINE
DO 159 I=1,KMAX
DO 159 K=1,M
CALL STR1(I,K,.FALSE.,T)
IF(AS(LI(I),K))CALL STR1(I,K,.TRUE.,T) @REORGANISATION DES ASS.UN.
159 CONTINUE
GO TO 162
158 CALL TRSF(AR,LT,G,E,F,H) @COMPLEMENTATION DE G K
C..... FUSION CONTROLEE PAR G.K DES ASS.UN. N'APPARTENANT PAS A [LJ]
KT=0
IF(APR)PRINT 500
AW=.FALSE.
DO 31 I=1,LT
DO 32 J=1,KMAX
IF(I.EQ.LI(J))GO TO 31 @K(I) APPARTIENT A [LJ]
32 CONTINUE
KT=KT+1
LS(KT)=I
31 CONTINUE
168 DO 151 I=1,KMAX
DO 151 K=1,LT
CALL STR1(I,K,.FALSE.,T) @REINITIALISATION DE R
151 CONTINUE
KL=0
DO 38 II=1,KT
I=LS(II)
DO 34 K=1,M
34 G(K)=AS(I,K)
MX=0
DO 33 J=1,KMAX
NE(J)=0
IF(AR(LI(J),I).OR.AR(I,LI(J)))GO TO 33 @LA J IEME ASS.UN. DE [LJ] EST POSITIE
C PAR RAPPORT A LA IEME ASS.UN. (PAS DE FUSION)
DO 35 K=1,M
35 H(K)=AS(LI(J),K)
CALL INTR(G,H,F,M)
NE(J)=ICARD(F,M)
MX=MAX(MX,NE(J))
33 CONTINUE
C..... LA VALEUR MAXIMALE DU CARDINAL DE L'INTERSECTION EST-ELLE UNIQUE
K=0
DO 46 J=1,KMAX
IF(NE(J).LT.MX)GO TO 46
K=K+1
ID=J
46 CONTINUE
IF(K.EQ.1)GO TO 37
KL=KL+1
NOA(KL)=I @MEMORISATION DES ASS.UN. N'AYANT PU FUSIONNER
GO TO 38
37 CALL STR1(ID,I,.TRUE.,T) @MEMORISATION DES ASS.UN. FUSIONNANT
PRINT 409,I,LI(ID)
409 FORMAT(1X,I3,' FUSIONNE AVEC',I2)
38 CONTINUE
C..... FUSION DES ASS.UN.
DO 152 I=1,KMAX
DO 154 J=1,M
154 G(J)=AS(LI(I),J) @CONTENU DE LA I EME ASS.UN. DE [LJ]
DO 155 J=1,LT
IF(.NOT.AT1(I,J))GO TO 155
DO 153 K=1,M
153 H(K)=AS(J,K) @CONTENU DE LA J EME ASS.UN.
CALL REUN(G,H,F,M) @FUSION DES DEUX ASS.UN.
DO 156 K=1,M
156 G(K)=F(K)
155 CONTINUE
DO 157 K=1,M
AS(LI(I),K)=.FALSE.
IF(G(K))AS(LI(I),K)=.TRUE.
157 CONTINUE
152 CONTINUE
IF(KL.EQ.0.OR.AW)GO TO 170 @FUSION ACHEVEE
KT=KL
DO 167 I=1,KT
LS(I)=NOA(I)
167 AW=.TRUE.
GO TO 168 @LES ASS.UN. RESIDUELLES PEUVENT ELLES FUSIONNER AVEC [LJ]
170 DO 165 I=1,KMAX
DO 165 K=1,M
AR(I,K)=AS(LI(I),K)
165 CONTINUE
162 CALL PRTN(110,110,AR,KMAX,M,' R A N G E C H A R T 1
* ')
CALL IDENT(AR,AS,NC,KMAX,M,NE,LI,LS,NOA,NOB,B,F,G,AI,.TRUE.)
STOP
END

```

TABLE Ie

```

C      FUNCTION AT(I, J)
C      -----
C      LECTURE DE L'ELEMENT I, J DE LA PSEUDO-MATRICE SYMETRIQUE AT
COMMON/MATRIX/T(8000)
COMMON/DAT/M, NBIT
LOGICAL T, AT
AT=. FALSE.
K=MIN(I, J)
IND=(K-1)*M-((K-1)**2+K-1)/2+MAX(I, J)
ID=1+(IND/NBIT)
L=MOD(IND, NBIT)
IF(L.EQ. 0)GO TO 1
IF(BITS(T(ID), L, 1).EQ. 1)AT=. TRUE.
RETURN
1 IF(BITS(T(ID-1), 36, 1).EQ. 1)AT=. TRUE.
RETURN
END

C      FUNCTION AT1(I, J)
C      -----
C      LECTURE DE L'ELEMENT I, J DE LA PSEUDO-MATRICE NON SYMMETRIQUE AT1
COMMON/MATRIX/T(8000)
COMMON/DAT/M, NBIT
LOGICAL T, AT1
AT1=. FALSE.
IND=(I-1)*M+J
ID=1+(IND/NBIT)
L=MOD(IND, NBIT)
IF(L.EQ. 0)GO TO 1
IF(BITS(T(ID), L, 1).EQ. 1)AT1=. TRUE.
RETURN
1 IF(BITS(T(ID-1), 36, 1).EQ. 1)AT1=. TRUE.
RETURN
END

C      SUBROUTINE COMPL(A, B, C, N)
C      -----
C      ... COMPLEMENTAIRE DE B PAR RAPPORT A A
DIMENSION A(N), B(N), C(N)
LOGICAL A, B, C
DO 1 I=1, N
1 C(I)=A(I).AND. NOT. B(I)
RETURN
END

C      SUBROUTINE COUNT(I, J, IA, NA, IK, L)
C      -----
C      STATISTIQUE DES ARCS IMPLIQUES DANS LES Z4, S3, ET S4
DIMENSION IA(3000), IAS(110, 110)
COMMON/MATRAS/AS(110, 110)
LOGICAL AS, L
EQUIVALENCE (IAS, AS)
I1=MIN(I, J)
I2=MAX(I, J)
IF(NA.LE. 3000)GO TO 10
PRINT 300
300 FORMAT(1X, 'DEPASSEMENT DE CAPACITE DES VECTEURS IARC IART')
STOP
10 IF(NA.EQ. 0)GO TO 2
IF(L)IK=IAS(I1, I2)
IF(.NOT. L)IK=IAS(I2, I1)
IF(IK.EQ. 0)GO TO 2
IA(IK*3)=IA(IK*3)+1
RETURN
2 IA(NA+1)=MIN(I, J)
IA(NA+2)=MAX(I, J)
IK=NA/3+1
IA(NA+3)=1
IF(L)IAS(I1, I2)=IK
IF(.NOT. L)IAS(I2, I1)=IK
NA=NA+3
RETURN
END

C      SUBROUTINE CYCLE(I, J, AR, N, NCY, PL, SL, LR)
C      -----
C      DETECTION DES CYCLES ET CES CIRCUITS SEMI-ORIENTES DU GRAPHE G
C      MEMORISATION DES ARCS ET DES ARETES IMPLIQUEES DANS CES CONFI-
C      GURATIONS ET CONSTRUCTION DU GRAPHE DES ARCS IMPLIQUES (AT)
IMPLICIT LOGICAL (A-H), LOGICAL (P), LOGICAL (R-T)
COMMON/TEST/IARC(3000), IART(3000), NARC, NART
COMMON/MATRIX/T(8000)
DIMENSION PI(110), PJ(110), SI(110), SJ(110), PC(110), SC(110), C(110),
*A(110), B(110), AR(110, 110), IA(110), IB(110), PL(110), SL(110), LR(110)
C      ... L ARETE (I-J) EST ELLE IMPLIQUEE DANS UN S3 ,UN S4 OU UN Z4
CALL GAMMA(AR, J, N, P, J, SJ) @ SUCCESSEUR ET PREDECESSEURS DE I ET J
CALL REUN(PL, P, J, PC, N)
CALL REUN(SL, SJ, SC, N)
CALL INTR(PC, SC, C, N)
IF(ICARD(C, N).EQ. 0)GO TO 2 @LE COUPLE (I-J) N APPARAIT PAS DANS UN S3
C      ... L ARETE I-J EST IMPLIQUEE DANS UN S3 : ADJONCTION DIFFEREE D ARETE
DO 1 K=1, N
IF(.NOT. C(K))GO TO 1
CALL COUNT(I, K, IARC, NARC, IK, .TRUE.) @STATISTIQUE DES ARCS IMPLIQUES DANS
LES S3

```

TABLE If

```

CALL COUNT(J, K, IARC, NARC, IL, .TRUE.)
CALL STR(IK, IL, .TRUE., T) @CONSTRUCTION DU GRAPHE DES ARCS IMPLIQUES
C PRINT 100, I, J, K
  LR(I)=LR(I)+1
  LR(J)=LR(J)+1
  LR(K)=LR(K)+1
  CALL COUNT(I, J, IART, NART, IL, .FALSE.) @STATISTIQUE DES ARETES IMPLIQUEES
1 CONTINUE
C 100 FORMAT(1X, 'S3 DETECTE ENTRE (' , I3, 1H-, I3, ') ET ' , I3)
  NCY=NCY+1
C..... DETECTION DES S4 IMPLIQUANT L ARETE I-J
  2 CALL COMPL(PC, C, A, N) @COMPLEMENTAIRE DE C PAR RAPPORT A PC
  CALL COMPL(SC, C, B, N)
  IF(ICARD(A, N)*ICARD(B, N).EQ.0)RETURN @ L ARETE I-J PAS IMPLIQ. DANS S4
C RECHERCHE DES ARETES ENTRE LES ELEMENTS DE A ET DE B
  CALL INDEX(A, I, N, IA, NA)
  CALL INDEX(B, I, N, IB, NB) @ELEMENTS NON NULS DES VECTEURS A ET B
  DO 4 KA=1, NA
    K=IA(KA)
  DO 5 KB=1, NB
    L=IB(KB)
    IF(.NOT.(AR(K, L).AND.AR(L, K)))GO TO 5 @PAS D ARETE ENTRE K ET L
  CALL GAMMA(AR, K, N, PI, SI)
  CALL GAMMA(AR, L, N, PJ, SJ)
  CALL REUN(PI, PJ, PC, N)
  CALL REUN(SI, SJ, SC, N)
  CALL INTR(PC, SC, C, N)
  IF(C(I).OR.C(J))GO TO 5 @IL S'AGIT D'UN S3
C..... STATISTIQUE DES ARCS IMPLIQUES DANS LE S4 OU Z4 DETECTE
  IF(AR(I, K).NEGV.AR(K, I))CALL COUNT(I, K, IARC, NARC, IK, .TRUE.)
  IF(AR(J, K).NEGV.AR(K, J))CALL COUNT(J, K, IARC, NARC, JK, .TRUE.)
  IF(AR(I, L).NEGV.AR(L, I))CALL COUNT(I, L, IARC, NARC, IL, .TRUE.)
  IF(AR(J, L).NEGV.AR(L, J))CALL COUNT(J, L, IARC, NARC, JL, .TRUE.)
C..... STATISTIQUE DES ARETES IMPLIQUEES
  CALL COUNT(I, J, IART, NART, I1, .FALSE.)
  CALL COUNT(K, L, IART, NART, I2, .FALSE.)
C..... CONSTRUCTION DU GRAPHE DES ARCS IMPLIQUES DANS Z4 ET S4
  CALL STR(IK, IL, .TRUE., T)
C PRINT 101, MIN(I, J), MAX(I, J), MIN(K, L), MAX(K, L)
  LR(I)=LR(I)+1
  LR(J)=LR(J)+1
  LR(K)=LR(K)+1
  LR(L)=LR(L)+1
C 101 FORMAT(40X, 'S4 OU Z4 DETECTE ENTRE (' , I3, 1H-, I3, ') ET (' , I3, 1H-, I3, '
C *')')
  NCY=NCY+1
  5 CONTINUE
  4 CONTINUE
  RETURN
  END

FUNCTION EQUAL(E, F, N)
-----
C
C LES ENSEMBLES [E] ET [F] SONT-ILS EQUIVALENTS ?
  DIMENSION E(N), F(N)
  LOGICAL E, F, EQUAL
  EQUAL=.FALSE.
  DO 1 I=1, N
    IF(E(I).NEGV.F(I))RETURN
  1 CONTINUE
  EQUAL=.TRUE.
  RETURN
  END

SUBROUTINE GAMMA(AR, I, N, PRED, SUCC)
-----
C
C RECHERCHE DES PREDECESSEURS ET DES SUCCESSEURS DU SOMMET I DANS G
  DIMENSION PRED(N), AR(110, 110), SUCC(N)
  LOGICAL AR, PRED, SUCC
  DO 1 J=1, N
    PRED(J)=AR(I, J).AND..NOT.AR(J, I)
  1 SUCC(J)=AR(J, I).AND..NOT.AR(I, J)
  RETURN
  END

FUNCTION ICARD(A, N)
-----
C
C CARDINAL DE L'ENSEMBLE [A]
  LOGICAL A(N)
  ICARD=0
  DO 1 I=1, N
    IF(A(I))ICARD=ICARD+1
  1 CONTINUE
  RETURN
  END

SUBROUTINE IDENT(AS, AR, NC, LT, M, NE, LI, LS, NOA, NOB, B, F, G, AI, TT)
-----
C
C..... IDENTIFICATION DES ASSOCIATIONS UNITAIRES DANS LES SECTIONS
  IMPLICIT LOGICAL (A-H), LOGICAL (R-T)
  DIMENSION NE(M), LI(M), LS(M), NOA(110), NOB(200), B(M), F(M), G(M)
  *, AS(110, 110), NSR(200), AI(72, 110), AR(110, 110)
  DO 94 I=1, LT
  DO 152 J=1, M

```

TABLE Ig

```

152 AI(I,J)= FALSE.
DO 94 J=1,LT
AR(I,J)= FALSE.
C) 94 CONTINUE
IDENTIFICATION DES ASS UN. DANS LES SECTIONS
REWIND 10
DO 60 K=1,NC
READ(10)MI, (NE(I), LI(I), LS(I), I=1, MI)
LN=10E+9
LX=-10. E+9
DO 61 I=1, MI
LN=MIN(LI(I), LN)
61 LX=MAX(LS(I), LX)
KB=0
KSR=0
DO 62 I=LN, LX @POUR CHAQUE NIVEAU I
DO 64 J=1, M
64 B(J)= FALSE. @INITIALISATION
DO 63 J=1, MI
IF(I GE LI(J) AND I LE LS(J))B(NE(J))= TRUE @CONTENU DU NIVEAU I
63 CONTINUE
KX=0
DO 65 L=1, LT @POUR CHAQUE ASS UN. L
DO 66 J=1, M
66 F(J)=AS(L, J)
CALL INTR(B, F, G, M)
NOA(L)=ICARD(G, M)
65 KX=MAX(NOA(L), KX)
C) LE CARDINAL DE L'INTERSECTION (CONTENU NIVEAU-ASS. UN) A-T-IL
C) UNE VALEUR MAXIMALE UNIQUE ?
KA=0
DO 57 L=1, LT
IF(KX GT NOA(L))GO TO 67
KA=KA+1
ID=L
67 CONTINUE
IF(KA GT 1)GO TO 62 @LE NIVEAU I N'EST PAS ATTRIBUABLE A UNE SEULE
C) ASS UN. -REUNION DU CONTENU DES NIVEAUX ATTRIBUABLES A UNE ASS UN.
DO 150 L=1, M
150 F(L)=AI(ID, L)
CALL REUN(F, B, G, M)
DO 151 L=1, M
151 AI(ID, L)=G(L)
C)
KB=KB+1
NSR(KB)=I
NOB(KB)=ID @MEMORISATION DE L' ASS UN. IDENTIFIEE
IF(KB EQ 1. OR TT)GO TO 62
IF(NOB(KB) EQ NOB(KB-1))KB=KB-1
62 CONTINUE
IF(TT)GO TO 69
IF(K EQ 1)PRINT 406
DO 68 I=1, KB
DO 68 J=1, KB
IF(I EQ J)GO TO 68
AR(NOB(J), NOB(I))= TRUE. @CONSTRUCTION DE LA MATR. ASSOC. A G K
68 CONTINUE
PRINT 405, K, (NOB(I), I=1, KB)
GO TO 60
69 LN=(KB+29)/30 @IMPRESSION DES NIVEAUX ATTRIBUES UNIVOQUEMENT
ME=0
PRINT 409, K
409 FORMAT(///, 1X, 'P R O F I L ', I3, //)
DO 70 I=1, LN
MS=ME+1
ME=MS+30
ME=MIN(ME, KB)
PRINT 407, (NSR(J), J=MS, ME)
70 PRINT 408, (NOB(J), J=MS, ME)
60 CONTINUE
407 FORMAT(1X, 'NIVEAU. ', 30I4)
408 FORMAT(1X, 'ASS. UN. ', 30I4)
405 FORMAT(1X, I8, 2X, (20I5, /, 11X))
406 FORMAT(///, 1X, ' PROFIL SEQUENCE DES ASS UN IDENTIFIEES', //)
CALL PRTN(72, 110, AI, LT, M, ' R A N G E C H A R T 2 ' )
DO 153 I=1, LT
DO 153 J=1, M
153 AI(I,J)= FALSE
RETURN
END

SUBROUTINE INDEX(A, I, N, IA, NA)
C) -----
C) RECHERCHE DES ELEMENTS NON NULS DU VECTEUR A
DIMENSION A(110), IA(110)
LOGICAL A
NA=0
DO 1 J=1, N
IF( NOT A(J))GO TO 1
NA=NA+1
IA(NA)=J
1 CONTINUE
RETURN
END

```

TABLE Ih

```

C      SUBROUTINE INIT(R, L, M)
C      -----
C      INITIALISATION DU VECTEUR R
C      DIMENSION R(8000)
C      LOGICAL R, L
C      DO 1 I=1, M
C      DO 1 J=1, M
C      CALL STR(I, J, L, R)
1 CONTINUE
  RETURN
  END

C      SUBROUTINE INTR(A, B, C, N)
C      -----
C      INTERSECTION DES ENSEMBLES [A] ET [B]
C      DIMENSION A(N), B(N), C(N)
C      LOGICAL A, B, C
C      DO 1 I=1, N
1 C(I)=A(I).AND. B(I)
  RETURN
  END

C      SUBROUTINE NEIB(AR, K, G, N)
C      -----
C      ENSEMBLE DES VOISINS DU SOMMET K DANS LE GRAPHE G
C      DIMENSION G(N), AR(110, 110)
C      LOGICAL AR, G
C      DO 1 I=1, N
1 G(I)=AR(I, K).AND. AR(K, I)
  RETURN
  END

C      SUBROUTINE PRTN(ID1, ID2, AR, L, K, TIT)
C      -----
C      IMPRESSION DES MATRICES ASSOCIEES AUX GRAPHS
C      DIMENSION H(110), AR(ID1, ID2)
C      COMMON/TTR/TITL(19)
C      CHARACTER TIT*38, S*3
C      LOGICAL AR, H
C      S=3H...
C      N=FLOAT(K/40. )+0. 975
C      NE=0
C      DO 10 M=1, N
C      NS=NE+1
C      NE=NS+39
C      IF (NE. GT. K) NE=K
C      PRINT 100, TIT, TITL
100 FORMAT(///2X, A38, 19A4, /)
C      PRINT 102, (J, J=NS, NE)
102 FORMAT(6X, 60I3)
C      PRINT 103, (S, J=NS, NE)
103 FORMAT(6X, 40A3)
C      DO 1 I=1, L
C      DO 2 J=NS, NE
2 H(J)=AR(I, J)
1 PRINT 101, I, (H(J), J=NS, NE)
101 FORMAT(2X, I2, 2H: , 40I3)
10 CONTINUE
  PRINT 104
104 FORMAT(////)
  RETURN
  END

C      SUBROUTINE REUN(A, B, C, N)
C      -----
C      REUNION DES ENSEMBLES [A] ET [B]
C      DIMENSION A(N), B(N), C(N)
C      LOGICAL A, B, C
C      DO 1 I=1, N
1 C(I)=A(I).OR. B(I)
  RETURN
  END

```

TABLE II

```

C      SUBROUTINE STR(I, J, L, R)
C      -----
C      CONCENTRATION D'UNE MATRICE X(I, J) LOGIQUE SYMMETRIQUE DANS LES BITS DU
C      VECTEUR R. LA PSEUDO-FONCTION "BITS" A LA SIGNIFICATION SUIVANTE:
C      BITS(R(I), K, 1)=L : ON INTRODUIT LE CONTENU DE L DANS LE K IEME BIT
C      DU I EME MOT DU VECTEUR R.
C      L=BITS(R(I), K, 1) : ON TRANSFERE LE CONTENU DU K IEME BIT DU I EME
C      MOT DU VECTEUR R DANS LA VARIABLE L (VOIR FONCTIONS AT, AT1)
C      LE NOMBRE DE BITS PAR MOT EST FIXE DANS LE COMMON "DAT".
C      CETTE PSEUDO-FONCTION PEUT ETRE SIMULEE PAR L'UTILISATION DE MASQUES
C      DEFINIS EN OCTAL ET PAR LA COMBINAISON DE PRODUITS ET DE SOMMES LOGIQUES
      DIMENSION R(8000)
      LOGICAL R, L
      COMMON/DAT/M, NBIT
      K=MIN(I, J)
      IND=(K-1)*M-((K-1)**2+K-1)/2+ MAX(I, J)
      ID=1+(IND/NBIT)
      IF(ID. LE. 8000)GO TO 2
      PRINT 100
100  FORMAT(//, 1X, 'DEPASSEMENT DE CAPACITE DANS LE VECTEUR T')
      STOP
      2  K=MOD(IND, NBIT)
      IF(K. EQ. 0)GO TO 1
      BITS(R(ID), K, 1)=L
      RETURN
      1  BITS(R(ID-1), 36, 1)=L
      RETURN
      END

```

```

C      SUBROUTINE STR1(I, J, L, R)
C      -----
C      CONCENTRATION D'UNE MATRICE X(I, J) LOGIQUE NON SYMMETRIQUE DANS R
C      (VOIR FONCTION AT, AT1 ET COMMENTAIRE DANS LA SOUSROUTINE STR)
      DIMENSION R(8000)
      LOGICAL R, L
      COMMON/DAT/M, NBIT
      IND=(I-1)*M+J
      ID=1+(IND/NBIT)
      K=MOD(IND, NBIT)
      IF(K. EQ. 0)GO TO 1
      BITS(R(ID), K, 1)=L
      RETURN
      1  BITS(R(ID-1), 36, 1)=L
      RETURN
      END

```

```

C      SUBROUTINE TRSF(AR, LT, G, E, F, H)
C      -----
C      ANALYSE DU GRAPHE G, K ET DETECTION DES CIRCUITS C(N)
C      IMPLICIT LOGICAL (A-H), LOGICAL (R-T)
C      DIMENSION G(110), E(110), F(110), H(110), AR(110, 110)
C      TRANSFORMATION DE G, K
98  DO 25 I=1, LT
      DO 27 J=1, LT
      27  G(J)=AR(I, J)
      38  IF(ICARD(G, LT). EQ. 0)GO TO 25
      DO 36 L=1, LT
      IF(G(L))AR(I, L)=. TRUE.
      36  E(L)=. FALSE.
      IF(. NOT. G(I))GO TO 20
      PRINT 100
100  FORMAT(//, 1X, 'PROCEDURE INTERROMPUE: G, K CONTIENT DES CIRCUITS')
      STOP
      20  DO 26 K=1, LT
      IF(. NOT. G(K))GO TO 26
      DO 29 L=1, LT
      29  F(L)=AR(K, L)
      CALL REUN(E, F, H, LT)
      DO 49 L=1, LT
      49  E(L)=H(L) @ENS. DES PREDEC. DE L'ENS. DES PREDEC.
      26  CONTINUE
      DO 39 L=1, LT
      39  G(L)=E(L)
      GO TO 38
      25  CONTINUE
      RETURN
      END

```

