UNIVERSITÉ D'ALGER

ECOLE NATIONALE POLYTECHNIQUE

DEPARTEMENT ÉCONOMIE

THESE DE FIN D'ETUDES

RECHERCHE DE L'ISOMORPHIE

entre deux

GRAPHES

Proposée par :
Mme SAUCIER
Professeur de Mathématiques

Etudiée par :
Mourad FEREDJ

1969 - 70

Université d'Alger

Ecole Nationale Polytechnique

Département Economie

Thèse de fin d'études

RECHERCHE DE L'ISOMORPHISME ENTRE DEUX GRAPHES

Sujet proposé par

Mme SAUCIER

Professeur de Mathématiques

Elève Ingénieur

Mourad FEREDJ

Année Universitaire 1969 - 1970

— • —

Je remercie Mme SAUCIER pour toute l'aide qu'elle n'a cessé de m'accorder ainsi que pour la documentation qu'elle a bien voulu mettre à ma disposition.

Je remercie également Monsieur BOUMAHRAT dont les conseils m'ont été d'un grand secours dans la phase programmation de mon travail.

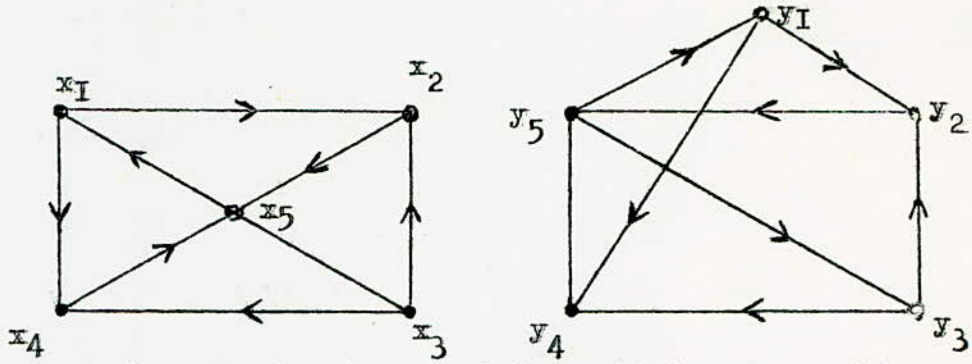
TABLE DES MATIERES

INTRODUCTION

Deux réseaux de communications peuvent avoir la même "structure". Considérons par exemple deux groupes de dix personnes et supposons que dans chaque groupe chacun peut communiquer directement avec tous les autres membres du groupe. Le graphe correspondant à ces deux réseaux de communications est le graphe complet symétrique (cf. Chap.I) à dix sommets. les deux groupes sont différents puisque composés de personnes différentes, mais leurs réseaux de communications ont la même structure. Supposons, pour donner un deuxième exemple que ces deux groupes soient organisés de la façon suivante : un Directeur ayant trois subordonnés, chacun de ces subordonnés ayant deux subordonnés. Ces deux groupes ont une même structure hiérarchique, constituée d'une source (d'autorité), de trois sommets ordinaires et des six puits.

Ces exemples illustrent le concept d'"isomorphisme" entre deux graphes. Deux graphes G_I et G_2 sont dits isomorphes s'il existe une correspondance biunivoque entre leurs sommets qui conserve les arcs. En d'autres termes G_I et G_2 sont isomorphes s'ils ont le même nombre des sommets et si l'on peut ordonner ces sommets ($x_I, \dots, x_p$ pour G_I, $y_I, \dots, y_p$ pour G_2) de façon à satisfaire la propriété suivante : quels que soient i et j, (x_i, x_j) existe dans G_I si et seulement si (y_i, y_j) existe dans G_2 . Une telle correspondance est appelee un isomorphisme entre G_I et G_2 .

Notons que bien souvent les dessins représentant deux graphes isomorphes G_I et G_2 ont des aspect différents; exemple: les graphes suivants :



Deux graphes sans boucle, complets, symètriques, ayant le même nombre de sommets sont nécessairement isomorphes. On peut donc affirmer qu'il y a un seul graphe sans boucle complet symètrique à cinq sommets, à un isomorphisme près, et parler " du " graphe sans boucle complet symètrique à cinq sommets.

De même on peut parler " du " graphe sans boucle à quatre sommets totalement disconnecté où chacun des quatre sommets est isolé, etc...

Etant donné deux graphes orientés, on se pose la question de savoir si ces deux graphes sont isomorphes ou non.

Il n'existe pas, à l'heure actuelle de solution algorithmique efficace à ce problème.

Si on indice les sommets avec le même ensemble d'indices, la méthode la plus naturelle consiste à permuter de toutes les façons possibles les sommets de l'un des graphes, et de le comparer à l'autre.

Malheureusement un tel algorithme est lent; il peut demander jusqu'à 40 ans de travail à un calculateur très rapide pour des graphes de I5 sommets seulement.

Donc tout de suite s'est imposée la nécessité de trouver un algorithme plus efficient.

Après avoir donné un aperçu sur le problème , nous allons donner , dans la première partie de cette étude , un exposé succint sur son aspect théorique et les principaux résultats nécessaires à sa compréhension ; pour plus de détails le lecteur pourra consulter la publication de J.P. STEEN (cf.biblio.) Une méthode de recherche d'un isomorphisme entre deux graphes sera donnée à titre d'exemple : la méthode d'UNGER a été choisie pour sa clarté et la simplicité de son application ; une méthode proposée sera ensuite exposée ; nous préciserons ultérieurement les limites de son application. Nous traiterons des exercices d'application à l'aide des deux méthodes dans le but de rendre plus concrets les exposés théoriques

Enfin , la méthode proposée a été programmée en langage FORTRAN

I. Etude théorique du problème

I.I. Définitions

I.I.I. Définitions générales de la théorie des graphes

I.I.I.I. Graphes

- Graphe : soit un ensemble X et une application de X dans X; on appelle graphe G et on note $G=(X,\Gamma)$ le couple constitué par l'ensemble X et l'application Γ.
- Sous-graphe de G : $G'=(X,\Gamma')$ où $\Gamma'x \subset \Gamma x \ \forall x \in X$

I.I.I.2. Arcs

- Arête : deux arcs de sens contraires entre deux sommets. On les représente par un arc non orienté.
- Arcs adjacents : ont une extrémité commune.
- Arc adjacent à un sommet : arc dont ce sommet est une des extrémités.
- Boucle : arc dont les extrémités sont confondues.

I.I.I.3. Sommets

- Demi-degré extérieur (resp. intérieur) d'un sommet : nombre d'arcs partant de (resp. arrivant à) ce sommet.
- Degré : somme des deux demi-degrés.
- Successeur d'un sommet : sommet extrémité d'un arc issu de ce sommet.
- Prédécesseur d'un sommet : sommet origine d'un arc aboutissant à ce sommet.
- Voisin : prédécesseur ou successeur.

I.I.I.4. Suites d'arcs

- Chaîne : suite d'arcs tels que chacun d'eux est adjacent au précedent. Sa longueur est le nombre d'arcs.
- Cycle : chaîne fermée.
- Cycle élémentaire : cycle ne passant pas deux fois par le même sommet.
- Chemin ou circuit élémentaire : ne passe pas deux fois par le même sommet.

I.I.I.5. Graphes spéciaux

- Graphe connexe ; entre deux sommets quelconques il existe une chaîne
- Composante connexe : sous-graphe connexe, non relié au reste du graphe.
- Composante fortement connexe : entre deux sommets quelconques il existe un chemin.
- Arbre : graphe connexe, sans cycles.
- Arborescence : arbre tel qu'en chaque sommet n'aboutit qu'un arc au plus.
- Graphe bipartite : dont les cycles sont de longueur paire.

I.I.2. Isomorphisme entre deux graphes

I.I.2.I. Définition

Deux graphes $G_I = (X_I, \Gamma_I)$ et $G_2 = (X_2, \Gamma_2)$ sont isomorphes s'il existe une correspondance biunivoque entre les ensembles X_I et X_2 qui soit compatible avec les applications Γ_I et Γ_2.

On déduit de cette définition que deux graphes isomorphes ont même nombre de sommets n.

En pratique on indexe les sommets des deux graphes à l'aide du même ensemble d'indices. À la correspondance biunivoque entre les sommets est associée une permutation des indices des sommets. On déduit de la correspondance entre les sommets une correspondance entre les arcs.

Isomorphisme :

Correspondance Biunivoque

Numérotation

Permut. ν :

$$
\begin{tikzcd}
 & G_I \arrow[r, leftrightarrow] & G_2 & & \\
\mu_x : & X_I \arrow[r, leftrightarrow] \arrow[d, leftrightarrow, "\alpha"'] & X_2 \arrow[r, Rightarrow] \arrow[d, leftrightarrow, "\alpha"'] & \mu_u : U_I \arrow[r] & U_2 \\
 & \{I,2,\ldots n\} \arrow[r, leftrightarrow] & \{I,2,\ldots n\} & &
\end{tikzcd}
$$

I.I.2.2. Deuxième définition

Appelons μ_x la correspondance biunivoque entre les sommets et μ_u la correspondance entre les arcs.

μ_u se définit à partir de μ_x par la formule :

$\forall u_I \in U_I : u_I = (x_I, y_I) \iff \mu_u u_I = (\mu_x x_I, \mu_x y_I)$

un isomorphisme sera un couple de correspondance $(\mu x, \mu u)$

I.2. Automorphismes de graphes

il peut exister plusieurs correspondances biunivoques entre les sommets de deux graphes. Leur étude se fait en utilisant les automorphismes de chacun des graphes.

I.2.I. Définitions

I.2.I.I. Définition I

Un automorphisme d'un graphe est un isomorphisme de ce graphe sur lui-même.

On constate que l'application biunivoque de la définition d'un isomorphisme est ici une permutation des sommets.

I.2.I.2. Définition 2

Soit un graphe $G = (X, U)$.(*) Un automorphisme σ de G est défini par une permutation ν_x^σ des sommets de X telle que : $\forall u \in U : u = (x, y) \quad \exists u' \in U \quad u' = (\nu_x^\sigma x, \nu_x^\sigma y) \quad (u' = \nu_u^\sigma u)$.

On dira d'une permutation ν_x qui vérifie la condition ci-dessus, qu'elle respecte la structure du graphe.

I.2.2. Groupe d'automorphismes

L'ensemble des permutations définies ci-dessus forme un groupe H qui est un sous-groupe du groupe S_n des permutations de $n = |X|$ objets. Comme à chacune d'elles est associé un automorphisme,

(*) L'ensemble des arcs U détermine complètement l'application Γ du graphe, tout comme l'application Γ détermine l'ensemble U; pour cette raison on peut écrire indifféremment le graphe sous la forme $G(X, \Gamma)$ ou $G = (X, U)$.

I.2.2.I. Proposition

L'ensemble des automorphismes d'un graphe G forme un groupeH(G) isomorphe au sous-groupe H.

Par exemple pour un graphe plein (entre 2 sommets quelconques il existe un arc dans chaque sens) et pour un graphe sans arcs (chaque sommet est isolé) les groupes d'automorphismes H (G) sont isomorphes à S_n car toutes les permutations des sommets respectent les arcs.

Un graphe G, son complémentaire (ils ont mêmes sommets et leurs ensembles d'arcs sont complémentaires dans $X \times X$ et son inverse (obtenu en inversant chaque arc) ont même groupe d'automorphismes.

Une permutation qui respecte les arcs dans un de ces 3 graphes, les respecte dans les autres.

I.2.3. Isomorphisme et automosphismes

Trois propriètés vont nous permettre de voir l'intérêt d'étudier les automorphismes.

I.2.3.I. Proposition

Si deux graphes sont isomorphes, leurs groupes d'automorphismes sont isomorphes.

I.2.3.2. Proposition

Tout isomorphisme entre deux graphes, se construit en faisant le produit d'un isomorphisme entre les deux graphes par un automorphisme quelconque de l'un des deux graphes, le produit se faisant à droite ou à gauche suivant que l'automorphisme est défini sur le graphe de départ ou sur le graphe d'arrivée de l'isomorphisme.

Conclusion

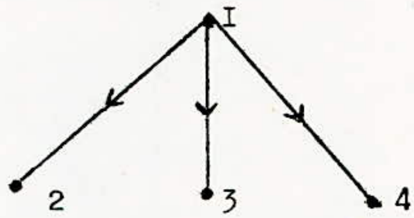
On peut construire, à partir d'un isomorphisme et du groupe d'automorphismes de l'un des graphes tous les isomorphismes qui existent entre les 2 graphes.

I.2.3.3. Proposition

Il existe une correspondance biunivoque entre les graphes isomorphes à un graphe donné G et les classes dans S_n du sous-groupe H isomorphe au groupe des automorphismes H (G) du graphe G à condition que tous ces graphes soient indicés par le même ensemble.

I.2.3.4. Exemple

soit un graphe G = (X,U)



X={I,2,3,4}
U={(I,2),(I,3),(I,4)}

On peut échanger les indices des sommets 2,3,4 sans modifier le graphe. Le groupe H (G) est isomorphe au sous groupe H du groupe S_4 des permutations de 4 objets qui laissent l'objet I invariant. H est isomorphe au groupe S_3 des permutations de 3 objets.

Les graphes isomorphes à ce graphe se déduisent de ce dernier par des permutations qui ne laissent pas le sommet I invariant. Comme on ne peut appliquer que les sommets 2,3,4 sur le sommet I, à toutes les permutations qui appliquent un de ces sommets sur le sommet I correspond un graphe isomorphe, dont le groupe d'automorphisme est isomorphe au sous-groupe des permutations de 4 objets qui laissent le sommet appliqué sur le sommet I invariant. Il existe trois graphes isomorphes à G, et 6 isomorphes pour chacun d'eux. Si on représente le graphe par I(2,3,4) on a :

a) Ses automorphismes qui appliquent G sur :
H(G)=(I(2,3,4),I(4,2,3),I(3,4,2),I(2,4,3),I(3,2,4),I(4,3,2)}
b) Ses isomorphismes qui appliquent le graphe sur :
$H(G_I)$={2(I,3,4),2(4,I,3),2(3,4,I),2(I,4,3),2(3,4,I),2(4,3,I)}
$H(G_2)$={3(2,I,4),3(4,2,I),3(I,4,2),3(2,4,I),3(I,2,4),3(4,I,2)}
$H(G_3)$={4(2,3,I),4(I,2,3),4(3,I,2),4(2,I,3),4(3,2,I),4(I,3,2)}

I.3. Equivalences

A l'aide du groupe d'automorphismes d'un graphe G on peut définir sur l'ensemble des sommets des relations d'équivalence.

I.3.I. H - Equivalence

Définition

Soit H le sous-groupe du groupe des permutations S_n isomorphe au groupe H (G) des automorphismes d'un graphe G (X,Γ) de n sommets.

Deux sommets x et y de X sont H-équivalent s'il existe une permutation ν de H telle que $y = \nu x$.

$$x \underset{H}{\sim} y \Leftrightarrow \exists \nu \in H \ : \ y = \nu x.$$

Alors la classe d'un sommet x pour cette relation d'équivalence sera définie par : $H_x = \{y/y \in X, \exists \nu \in H : y = \nu x\} = H.x.$

Il faut connaître le groupe H pour calculer la H - partition de X associée à cette équivalence.

I;3.2. S-équivalence

I.3.2.I. Définition

Deux sommets x et y de X sont S-équivalents si la transposition τ_{xy} appartient à H.

$$x \underset{S}{\sim} y \Leftrightarrow \tau_{xy} \in H.$$

On définit ainsi une relation d'équivalence : la S- classe S_x d'un sommet x sera: $S_x = (y/y \in X, \ y = \tau_{xy}, \ \tau_{xy} \in H).$

De la définition des automorphismes, on déduit :

I.3.2.2. Proposition

τ_{xy} appartient à H signifie que x et y ont :

- mêmes successeurs,
- mêmes prédécesseurs,
- et sont ou ne sont pas, tous les deux, supports d'une boucle.

I.3.3. Comparaison des équivalences

Proposition

la S-équivalence implique la H-équivalence mais non l'inverse.

Autrement dit la S-partition est plus fine que la H-partition.

En effet si x est S-équivalent à y, il existe $\nu = \tau_{xy} \in H$ telle que $y = \nu x$, d'où x est H-équivalent à y. L'inverse n'est pas vrai, même si les permutations de H se décomposent en produits de transpositions, ces transpositions n'étant pas nécessairement des éléments de H.

I.3.4. Sommets invariants par H

I.3.4.I. Définition

I_H est l'ensemble des sommets invariants par H.

$$I_H = (x / x \in X, \forall \nu \in H \quad \nu x = x)$$

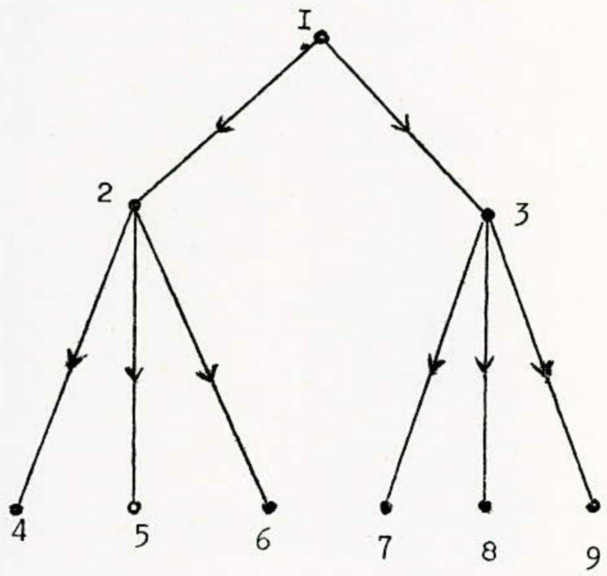
I.3.4.2 Proposition

chaque sommet de I_H est une H-classe est une S-classe.

En effet H. x = x pour tout $x \in I_H$ et $\tau_{xx} = I$ pour chacun d'eux.

I.3.5. Exemple

$I_H : \{I\}$



S-classes :
(4,5,6) et
(7,8,9)

Deux S classes non réduites à I sommet.

Deux H-classes non réduites à I sommet

$I_H = \{I\}$

Le groupe H peut se représenter par :

$H = S_3$ (sur 4,5,6) x S_3 (7,8,9) $\times r_I$ où $r_I = [2,3][4,7][5,8][6,9]$

Notation : $[a,b]$ est un cycle de permutation.

I.4 Conditions nécessaires d'isomorphisme

Une condition suffisante permettrait de dire si deux graphes sont isomorphes. Si une condition nécessaire n'est par vérifiée, alors les deux graphes ne sont pas isomorphes.

Comme on ne connaît pas actuellement de condition suffisante simple, toutes les méthodes de recherche de l'isomorphisme sont basées sur les conditions nécessaires d'isomorphisme qui ont l'avantage d'être faciles à établir et à vérifier.

I.4.I. Degré des sommets correspondants

I.4.I.I. Proposition

Les sommets correspondants de deux graphes isomorphes ont même demi-degré interieur et exterieur, et sont support du même nombre de boucles.

I.4.I.2. Corollaire

Deux graphes isomorphes ont même nombre de sommets de même demi-degré, et même nombre de sommets support du même nombre de boucles.

I.4.2. Longueur des circuits

I.4.2.I. Proposition

Pour deux graphes isomorphes, si deux arcs sont adjacents dans l'un des graphes, leurs correspondants le sont aussi dans l'autre graphe (arrivant, partant ou se suivant en un sommet).

I.4.2.2. Proposition

L'isomorphisme entre deux graphes induit une correspondance biunivoque entre les cycles et entre les circuits

I.4.2.3. Corollaire I

Si deux graphes sont isomorphes, ils ont même nombre de cycles et de circuits de même longueur.

I.4.2.4. Corollaire 2

Pour deux graphes isomorphes, par deux sommets correspondants il passe le même nombre de circuits et cycles de même longueur.

I.4.3. Les relations d'équivalence

Si deux graphes sont isomorphes, les relations d'équivalence entre les sommets doivent être respectées, à condition que celles-ci soient définies directement ou indirectement à l'aide des arcs.

Par exemple : $x \sim y \Leftrightarrow d^o(x)=d^o(y)$, donc

I.4.3.I. Proposition

Un isomorphisme respecte les équivalences entre les sommets

I.4.3.2. Corollaire

Si deux graphes sont isomorphes, il ont, pour toute relation d'équivalence du type I.4.3., même nombre de classes de même cardinal.

I.4.4. Les propriétés des graphes

De la définition on déduit :

Proposition

Si deux graphes sont isomorphes, ils sont ensemble symètriques ou non symètriques.

2. Exemple d'une méthode de Recherche de l'isomorphie :

Méthode d' UNGER

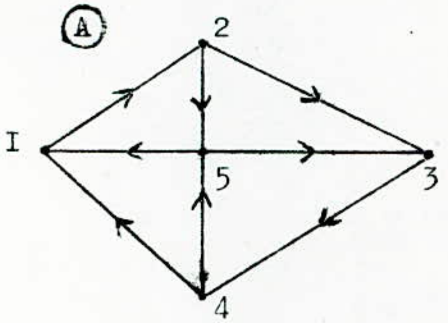
Nous allons commencer par donner un exposé de la méthode dans lequel nous essaierons surtout d'expliquer le contenu de l'organigramme.

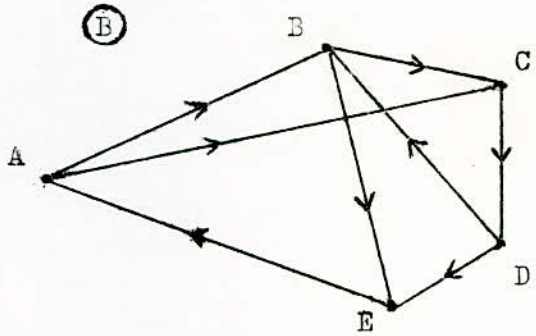
Ensuite suivront 3 exercices d'application.

2.I. Exposé de la méthode

Soient 2 graphes A et B (ayant le même nombre de sommets). On forme une liste de correspondance de l'ensemble des sommets de A avec l'ensemble de ceux de B. Les IDD de chaque sommet sont alors trouvés et servent à partitionner les ensembles en ceux dont les IDD sont égaux. La liste de correspondance est appelée PNPL.

Prenons l'exemple suivant :





Le PNPL initial est :

I,2,3,4,5 (A,B,C,D,E).

Après avoir opéré une partition sur la base des IDD, le PNPL devient :

I,3,5 (B,C,E) 2,4 (A,D).

Noter que les ensembles correspondants pour les graphes A et B doivent avoir le même nombre d'éléments à chaque étape si les graphes sont isomorphes. Le PNPL précédent indique, entre autres choses, que les sommets 5 et B doivent correspondre.

Une fonction est évaluée pour les deux graphes , à la suite de quoi le PNPL est mis à jour (en affinant les partitions). Le processus est répété pour une serie de fonctions . A des intervalles réguliers, un choix (hypothése) est fait sur un couplage possible, basé sur le dernier PNPL obtenu , puis on fait un test de validité. Dans notre exemple une hypothése aurait pu être faite: le graphe A pourrait être transformé en B au moyen de l'ensemble de transformation : I-->C, 3--> E, 5--> B, 2--> A, 4--> D. Aprés avoir éffectué la transformation de A et comparé le graphe obtenu à B, on devrait voir que l'hypothése faite n'était pas valide. On devrait alors procéder à d'autres essais pour affiner le PNPL.

Des fonctions additionnelles peuvent être calculées avec efficience. Citons:

a: le nombre de sommets qui peuvent être atteints à partir du sommet i par un chemin de longueur n (une séquence de fonctions est ainsi définie).

b: le nombre de sommets à partir desquels i peut être atteint par un chemin de longueur n,

c: une fontion à valeurs binaires égale à I sur les sommets inclus dans des circuits de longueur n, et

d: plusieurs variations de chacune de celles-ci.

Une autre procédure importante pour générer des fonctions nodales (des sommets) est appelée Extenting (cf plus loin).

Le processus d'affinement du PNPL par le calcul des fonctions nodales peut s'achever de l'une des trois manières suivantes:

I: Un choix s'avere correct, auquel cas le programme se termine par l'exhibition de la transformation donnant l'isomorphisme.

2: Un PNPL non valide est produit; cela étant, on trouve que pour un ensemble de n sommets de A équivalents en ce qui concerne toutes les propriétés testées pour un point donné, l'ensemble des sommets de B correspondants contient n_B éléments où $n_A \neq n_B$. Ceci signifie que les graphes ne peuvent être isomorphes.

3: Ni l'une ni l'autre des 2 situations précédentes n'a lieu et on détermine (d'une manière qui sera discutée plus tard) que cette ligne d'attaque doit être abandonnée.

Dans le dernier cas, un programme est alors éxécuté qui consiste à faire des hypothéses en vue d'affiner le PNPL, en changeant ces hypothéses quand certaines déductions des PNPL obtenus s'avérent non valides, et en faisant encore des hypothése occasionnelles sur la base du dernier PNPL. Ce programme conduit éventuellement à une solution.

Exemple:

soient 2 graphes A et B:

A : I(3,4,5)2(I,4,6)3(2,6)4(2,3)5(4,)6(I,3,5)

B : I(2,4,6)2(4,5,6)5(I,2,5)4(I,3)5(3,4)6(5)

(les arcs de A sont : (I,3),(I,4),(I,5),(2,I),(2,4) etc.)

Sur la base de la fonction ODD, nous obtenons le PNPL I,2,6 (I,2,3)3,4(4,5)5(6). La fonction IDD n'améliere pas la partition .

Les couples d'ensembles de sommets tels que (I,2, 6) et (I,2,3) ci-dessus peuvent être considérés comme des set pairs (couples d'ensemble) du PNPL. Un tel set pair consiste en un ensemble de sommets du graphe A, sur lesquels toutes les fonctions nodales évaluées jusqu'ici ont les mêmes valeurs , et l'ensemble correspondant de sommets du graphe B. Les autres set pairs ci-dessus sont (3,4), (4,5) et (5,6).

Supposons maintenant qu'on affecte un nombre unique (appelé set number(nombre d'ensemble)) à chaque set pair du PNPL ci-dessus. Les set numbers peuvent être affectés séquenciellement aux set pairs.

Dans l'exemple présent,I,2, et 3 sont affectés respectivement aux trois set pairs. Affectons maintenant un node number (nombre nodal) égal au set number du set pair auquel il appartient. Ainsi , dans notre exemple , on affecte aux sommets I à 6 de A les nombres I,I,2,2,3, et I respectivement.

Pour evaluer pour le $i^{ème}$ sommet la fonction nodale "EXTEND" sur les graphes originaux on additionne les node numbers correspondant aux successeurs du sommet i.

Ainsi dans notre exemple, les successeurs du sommet I. de A sont 3,4 et 5 ayant les node numbers 2,2, et 3 respectivement. D'où : la fonction Extend affecte la valeur 7 au sommet I de A.

Cette fonction est suffisante pour partitionner le PNPL en des ensembles dont les membres sont réduits à un seul élément.

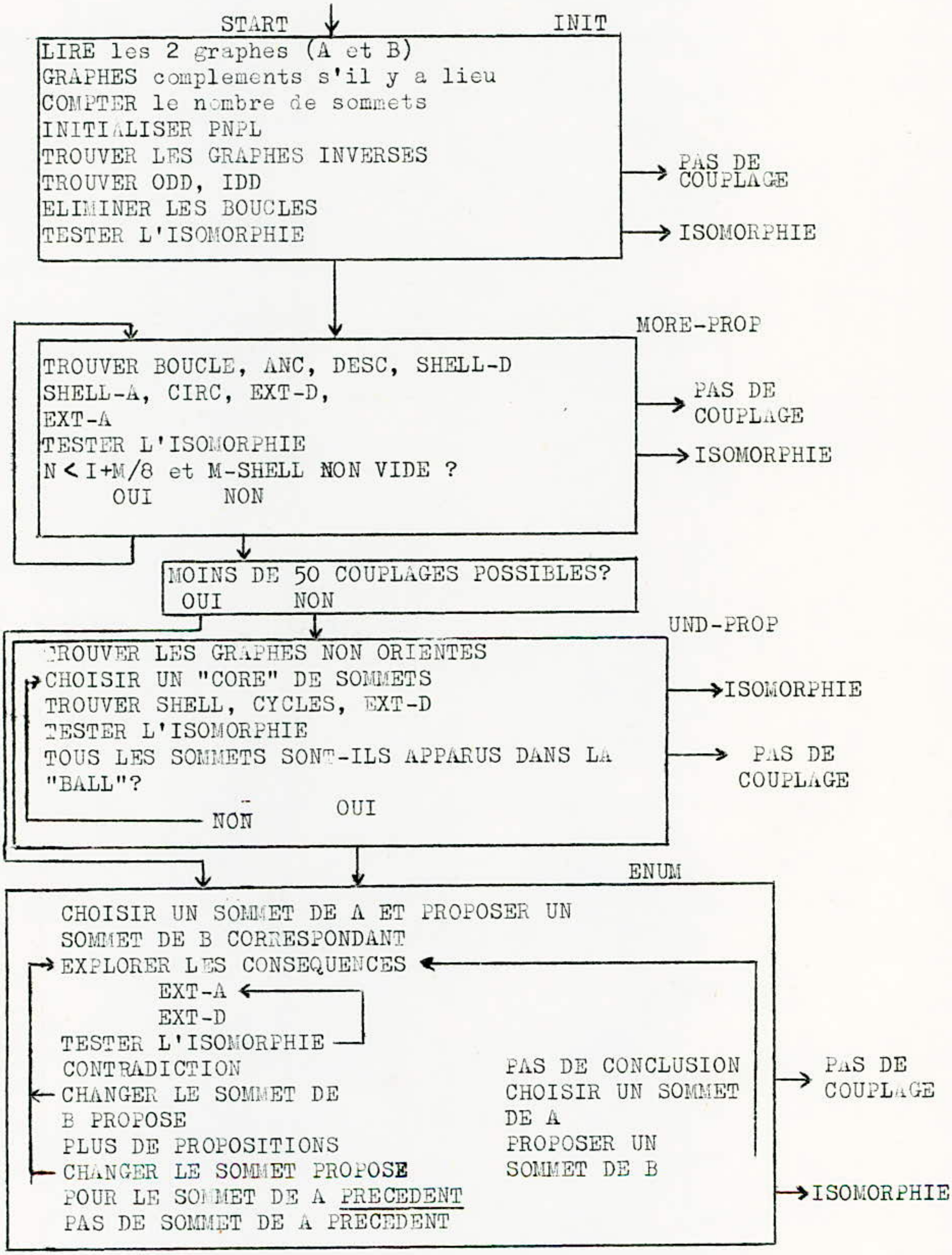
I(2)2(3)3(4)4(5)5(6)6(I). Si ceci n'est pas le cas , le nouveau PNPL peut être utilisé à évaluer une autre fonction, ce qui a peur effet d'affiner la partition.

Ce processus peut être répété jusqu'à ce qu'un couplage soit trouvé et vérifié, une contradiction obtenue, ou aucun changement n'affecte le PNPL.

Au lieu d'utiliser les graphes originaux (listes de successeurs) pour la fonction extend nous pouvons utiliser les listes des descendants ou ancêtres de la $n^{\text{ème}}$ - génération.

Dans l'exemple ci-dessus , la fonction extend affecte le nombre 2 aux sommets3 et 5 du graphe A(et 4 et 6 de B) en sommant différents ensembles de nodes numbers (I+I pour le sommet 3 et juste 2 pour le sommet 5). Bien que rien ne soit perdu dans ce cas , car 3 et 5 étaient initialement dans des ensembles (set pairs) de PNPL différents, en général cet effet réduit légérement la discrimination de la fonction extend.

Organigramme



Notations:

IDD: Inward demi-degree (demi-degré intérieur)

ODD: outivard " (" extérieur)

PNPL: possible node pairing list (liste des couplages possibles de sommet).

Ornigramme

ANC : ancester (ancêtre)

DESC: descendant

SHELL: n-shell(n-écaille): le n-shell d'un sommet i est constitué par l'ensemble des sommets qui peuvnet être atteints à partir de i car un chemin de longueur n, et qui ne peuvent être atteints à partir de i par aucun chemin plus court.

SHELL-D : n shell - descendants (n-écaille des descendants).

SHELL-A : " - ancestors (" ancêtres)

CIRC : n-circuits de longueur n)

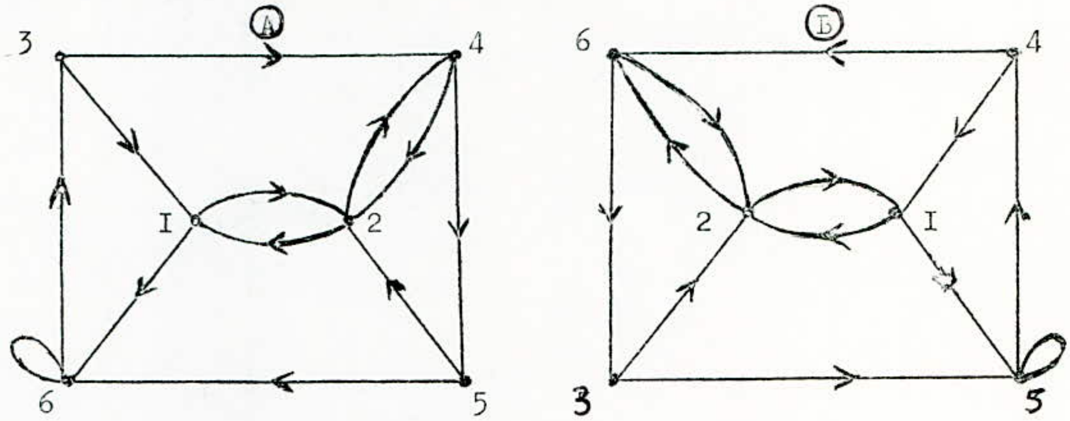
CORE : (noyau)

BALL : n-ball (n-ball): ensemble des sommets appartenant à l'union de tous les K-shells pour le n.

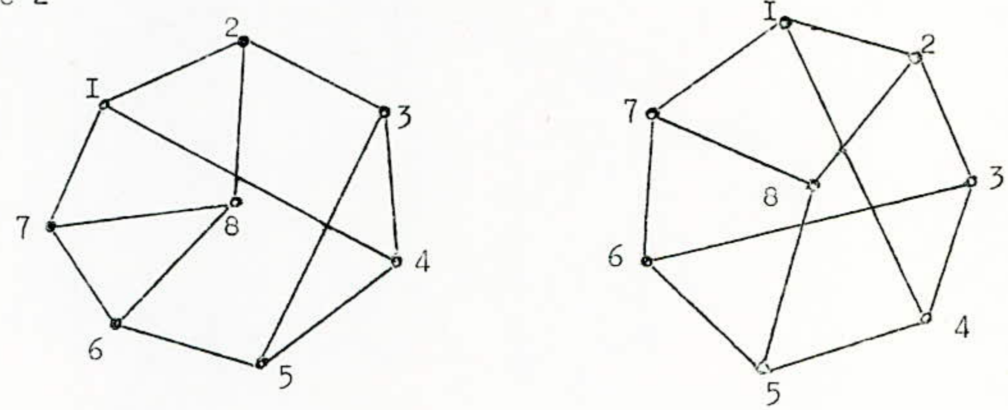
La n-ball d'un sommet i est l'ensemble des sommets q ui peuvent être atteints à partir de i par un chemin quelconque de longueur n.

2.2. EXERCICES : Recherche d'un isomorphisme entre 2 graphes

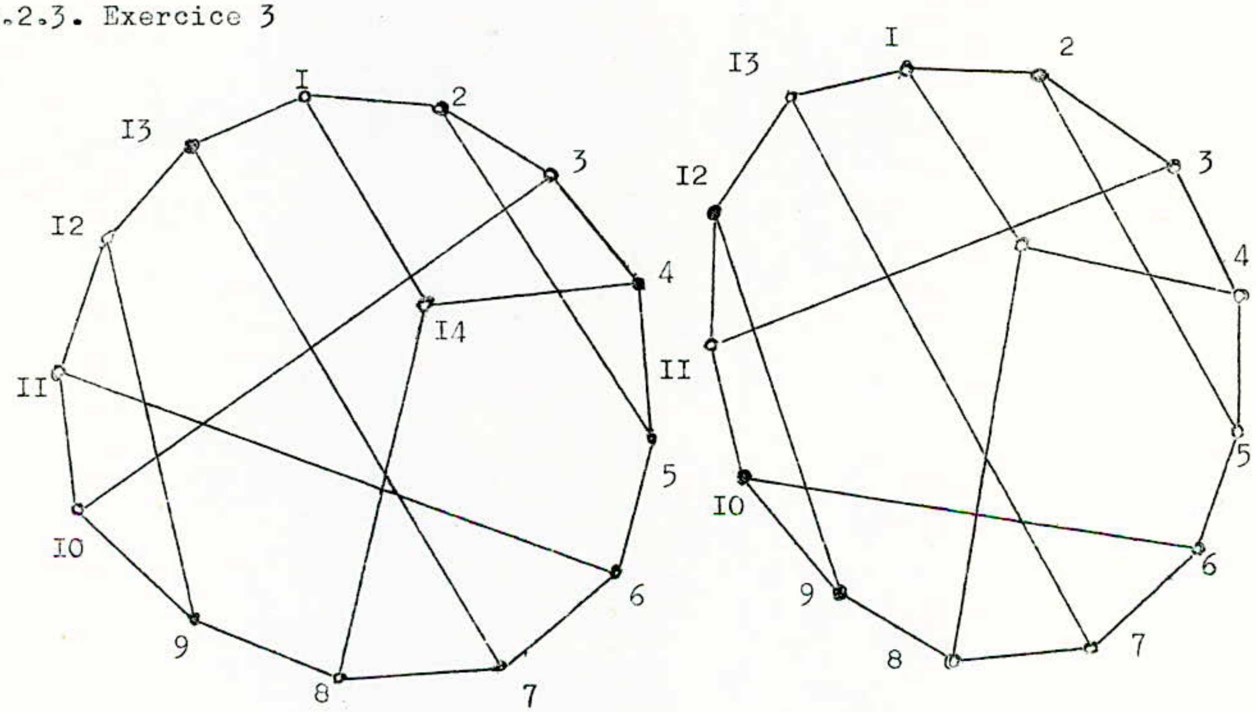
2.2.I. Exercice I



2.2.2. Exercice 2



2.2.3. Exercice 3



Résolution par la méthode d'UNGER

2.2.I . Exeercice I

G_I: I(2,6)2(I,4)3(I,4)4(2,5)5 (2,6)6(3,6)

G_2: I(2,6)2(I,5)3(I,5)4(2,6)5(4,5)6(I,3)

PNPL: I,2,3,4,5,6(I,2,3,4,5,6)

R(G_I): I (2,3)2(I,4,5)3(6)4(2,3)5(4)6(I,5,6)

R(G_2): I (2,3,6)2(I,4)3(6)4(5 5(2,3,5)6(I,4)

ODD--->PNPL: Inchangé

IDD--->PNPL: 2,6(I,5)I,4(2,6)3,5(3,4)

Elimination des boucles:---> 6(5)

PNPL: 6(5)2(I) I,4(2,6)3,5(3,4)

Test: pas de couplage.

More-prop

n = I

N N G_I: I(3)2(2)3(4)4(3)5(4)6(I)

G_2: I(2)2(3)3(4)4(4)5(I)6(3)

EXT-D G_I: I(3)2(6)3(6)4(6)5(3)6(4)

G_2: I(6)2(3)3(3)4(6)5(4)6(6)

PNPL: 6(5)2(I)I(2)4(6)3(4)5(3)

Résultat: isomorphisme

G_I:	I	2	3	4	5	6
G_2:	2	I	4	6	3	5

2.2.2. Exercice 2

G_I et R (G_I):

I(2,4,7)2(I,3,8)3(2,4,5)4(I,3,5)5(3,4,6)6(5,7,8)7(I,6,8)8(2,6,7)

G_2 et R(G_2):

I(2,4,7)2(I,3,8)3(2,4,6)4(I,3,5)5(4,6,8)6(3,5,7)7(I,6,8)8(2,5,7)

PNPL:I,2,3,4,5,6,7,8(I,2,3,4,5,6,7,8)

ODD et IDD---> PNPL inchangé.

Test : pas de couplage.

MORE PROP

n = I

Ext---> PNPL inchangé.

n = 2

A(G_I) et D (G_I): I (5)2(5)3(6)4(6)5(7)6(7)7(6)8(6)

A(G_2) et D (G_2): I(5) 2(5)3(5)4(5)5(5)6(5)7(5)8(5)

PNPL: I,2(I,2,3,4,5,6,7,8)3,4,7,8()5,6 ()

Pas de couplage possible.

Résultat: non -isomorphie.

2.1.3. Exercice 3

G_I et R(G_I):
I(2,I3,I4)2(I,3,5)3(2,4,IO)4(3,5,I4)5(2,4,6)6(5,7,II)7(6,8,I3)
8(7,9,I4)9(8,IO,I2)IO(3,9,II)II(6,IO,I2)I2(9,II,I3)I3(I,7,I2)
I4(I,4,8).

G_2 et R(G_2):
I(2,I3,I4)2(I,3,5)3(2,4,II)4(3,5,I4)5(2,4,6)6(5,7,IO)7(6,8,I3)8(7,9,I4)
9(8,IO,I2)IO(6,9,II)II(3,IO,I2)I2(9,II,I3)I3(I,7,I2)I4(I,4,8)

PNPL : I,2,3,4,5,6,7,8,9,IO,II,I2,I3,I4(I,2,3,4,5,6,7,8,9,IO,II,I2,I3,
I4).

ODD ----> PNPL inchangé

IDD ----> " "

Test: pas de couplage.

MORE-PROP

n=I

EXT ----> PNPL inchangé

n=2

A(G_I) et D(G_I) :
I(7)2(6)3(6)4(6)5(6)6(7)7(7)8(7)9(6)IO(6)II(6)I2(6)I3(7)I4(7)

A(G_2) et D(G_2) :
I(7)2(6)3(6)4(6)5(6)6(7)7(7)8(7)9(6)IO(6)II(6)I2(6)I3(7)I4(7)

PNPL : I,6,7,8,I3,I4,I,6,7,8,I3,I4)2,3,4,5,9,IO,II,I2(2,3,4,5,9,IO,II
I2)

S-A(G_I) et S-D(G_I) :
I(6)2(5)3(5)4(5)5(5)6(6)7(6)8(6)9(6)IO(6)II(6)I2(6)I3(6)I4(6)

S-A(G_2 et S-D(G_2) :
I(6)2(5) -- I4(6)

PNPL inchangé

Circuit ----> PNPL inchangé

EXT

N N

G_I : I(I)2(2)3(2)4(2)5(2)6(I)7(I)8(I)9(2)IO(2)II(2)I2(2)I3(I)I4(I)

G_2 : I(I)2(2)3(2)4(2)5(2)6(I)7(I)8(I)9(2)IO(2)II(2)I2(2)I3(I)I4(I)

EXT-A(G_I) et EXT-D(G_I) :

I(4)2(5)3(6)4(5)5(5)6(5)7(3)8(4)9(5)IO(6)II(5)I2(5)I3(4)I4(4)

EXT-A(G_2) et EXT-D(G_2) :

I(4)2(5)3(6)4(5)5(5)6(5)7(3)8(4)9(5)IO(5)II(6)I2(5)I3(4)I4(4)

PNPL : I,8,I3,I4(I,8,I3,I4)2,4,5,6,9,II,I2(2,4,5,6,9,IO,I2)3,IO(3,II)7(7)

n=3

A(G_I) et D(G_I) :

I(I2)2(IO)3(9)4(9)5(9)6(9)7(II)8(IO)9(9)IO(8)II(9)I2(IO)I3(I2)I4(II)

A(G_2) et D(G_2) :

I(II)2(9)3(9)4(IO)5(9)6(9)7(II)8(I2)9(IO)IO(9)II(8)I2(9)I3(IO)I4(I2)

PNPL : I,I3(8,I4)8(I3)I4(I)2,I2(4,9)4,5,6,9,II(2,5,6,IO,I2)3(3)IO(II)7(7)

N N

G_I : I(I)2(4)3(6)4(5)5(5)6(5)7(8)8(2)9(5)IO(7)II(5)I2(4)I3(I)I4(3)

G_2 : I(3)2(5)3(6)4(4)5(5)6(5)7(8)8(I)9(4)IO(5)II(7)I2(5)I3(2)I4(I)

EXT-A(G_I) et EXT-D(G_I) :

I(8)2(I2)3(I6)4(I4)5(I4)6(I8)7(8)8(I6)9(I3)IO(I6)II(I6)I2(II)I3(I3)I4(8)

EXT-A(G_2) et EXT-D(G_2) :

I(8)2(I4)3(I6)4(I2)5(I4)6(I8)7(8)8(I3)9(II)IO(I6)II(I6)I2(I3)I3(I6)I4(8)

PNPL : I(I4)I3(8)8(I3)I4(I)2(4)I2(9)4,5(2,5)6(6)9(I2)II(IO)3(3)IO(II)7(7)

S-A(G_I) et S-D(G_I) :

I(4)2(5)...4(4)5(4)......

S-A(G_2) et S-D(G_2) :

...2(4).........5(4).....

PNPL inchangé

ENUM Hypothèse:

On choisit : 4(2)5(5)

PNPL : I(I4)2(4)3(3)4(2)5(5)6(6)7(7)8(I3)9(I2)IO(II)II(IO)I2(9)I3(8)I4(I)

N N

G_I : I(I)2(2)3(3)4(4)5(5)6(6)7(7)8(8)9(9)IO(IO)II(II)I2(I2)I3(I3)I4(I4)

G_2 : I(I4)2(4)3(3)4(2)5(5)6(6)7(7)8(I3)9(I2)IO(II)II(IO)I2(9)I3(8)I4(I)

EXT-A(G_I) et EXT-D(G_I) :

I(29)......4(22)5(I2).......................................I4(I3)

EXT-A(G_2) et EXT-D(G_2) :

I(I3)2(22)...........5(I2)......................................I4(29)

Résultat : couplage --→ isomorphisme

G_I :	I	2	3	4	5	6	7	8	9	I0	II	I2	I3	I4
G_2 :	I4	4	3	2	5	6	7	I3	I2	II	I0	9	8	I

Remarque :

Si notre choix avait été :4(5)5(2),on aurait eu:

PNPL : I(I4)2(4)3(3)4(5)5(2)6(6)7(7)8(I3)9(I2)I0(II)II(I0)I2(9)I3(8)I4(I)

N N

G_I : I(I)2(2)3(3)4(4)5(5)6(6)7(7)8(8)9(9)I0(I0)II(II)I2(I2)I3(I3)I4(I4)

G_2 : I(I4)2(5)3(3)4(2)5(4)6(6)7(7)8(I3)9(I2)I0(II)II(I0)I2(9)I3(8)I4(8)

EXT-A(G_I) et EXT-D(G_I) :

I(29)....................4(22)5(I2)...........

EXT-A(G_2) et EXT-D(G_2) :

I(29)2(I3).....................5(I5)..........

Contradiction : les valeurs de la fonction Extend sur les sommets 4 de G_I et 5 de G_2 d'une part,5 de G_I et 2 de G_2 d'autre part ne sont pas égales.

On modifie notre choix : 4(2)5(5), et on retrouve ainsi le couplage précédent.

Notations :

- PNPL : cf. Not. 2.I.
- R(G) : Reverse graph (graphe inverse de G)
- N N : Node numbers (nombres nodaux)
- EXT-D : Ext-descendants (application de la fonction Extend aux descendants)
- EXT-A : Ext-ancestors (application de Extend aux ancêtres)
- A(G) : Ancêtres de G
- D(G) : Descendants de G
- S-A(G) : Shell-Ancestors (Ecaille des ancêtres de G)
- S-D(G) : Shell-Descendants (Ecaille des descendants de G)

3. Méthode de recherche de l'isomorphie entre 2 graphes

Pour des raisons de simplicité, nous exposons ces résultats dans un premier temps pour des graphes connexes, non orientés et bipartites. Ces résultats seront ensuite généralisés dans un deuxième temps.

Dans la recherche d'un isomorphisme entre 2 graphes, nous chercherons d'abord les automorphismes de ces graphes; nous verrons que le problème de la recherche des isomorphismes est alors souvent résolu.

3.I. Isomorphisme entre 2 graphes connexes, non orientés, bipartites.

Soient 2 graphes $G(X,\Gamma)$, $G'(X',\Gamma')$ connexes, non orientés et bipartites. Supposons qu'il existe un sommet $S \in X$, un sommet $S' \in X'$ tels que S corresponde à S' dans tout isomorphisme, s'il existe, entre G et G'. Autrement dit, nous supposons connue, dans un premier temps, une correspondance entre un sommet de G et un sommet de G'. Ayant distingué un sommet S dans G, nous étudions :

3.I.I. Automorphismes de G laissant un sommet S invariant

Le sous-groupe de H ainsi défini sera noté H_S et l'équivalence correspondante H_S- équivalence.

3.I.I.I. Structure en couches

Associons à $G(X,\Gamma)$ sa représentation en couches $(C_0, \ldots C_i \ldots)$ telle que

$S \in C_0, x \in C_i \iff d(S,x)=i$

nous avons les propriétés suivantes :

3.I.I.2. Propriété I

Soient $x, x' \in X$, si x correspond à x' par un automorphisme de H, alors x et x' appartiennent à une même couche.

3.I.I.3. Propriété 2

Orientons la structure en couches de C_0 vers C_I, $C_2 \ldots$ et appelons cette fois Γ^* la relation de succession du graphe orienté obtenu.

Γ^*_x est alors l'ensemble des successeurs de x dans la couche de rang supérieur.

$$x \longleftrightarrow x' \Longrightarrow \begin{array}{ccc} \Gamma_x^{*} & \longleftrightarrow & \Gamma_{x'}^{*} \\ \Gamma_x^{*2} & \longleftrightarrow & \Gamma_{x'}^{*2} \\ & \vdots & \\ \hat{\Gamma}_x^{*} & \longleftrightarrow & \hat{\Gamma}_{x'}^{*} \\ \Gamma_x^{*-1} & \longleftrightarrow & \hat{\Gamma}_{x'}^{*-1} \\ & \vdots & \\ \hat{\Gamma}_x^{*-1} & \longleftrightarrow & \hat{\Gamma}_{x'}^{*-1} \end{array}$$

Plus précisement, il découle de la définition que;

$$\left\{ \underset{h \in H_S}{x \longleftrightarrow x'} \right\} \Longleftrightarrow \left\{ \hat{\Gamma}_x^{*} \longleftrightarrow \hat{\Gamma}_{x'}^{*} \ \underline{\text{et}} \ \hat{\Gamma}_x^{*-1} \longleftrightarrow \hat{\Gamma}_{x'}^{*-1} \right\}$$

ç.à.d. 2 éléments se correspondent pas un automorphisme de H_S s'il existe une correspondance biunivoque entre l'ensemble de leurs descendants et de leurs ancêtres.

3.I.2. Algorithme d'affinement de partitions initiales

Etant donné une partition initiale, nous proposons une méthode rapide permettant d'affiner cette partition par étude des couches voisines successives.

3.I.2.I. Partition initiale des sommets d'une couche.

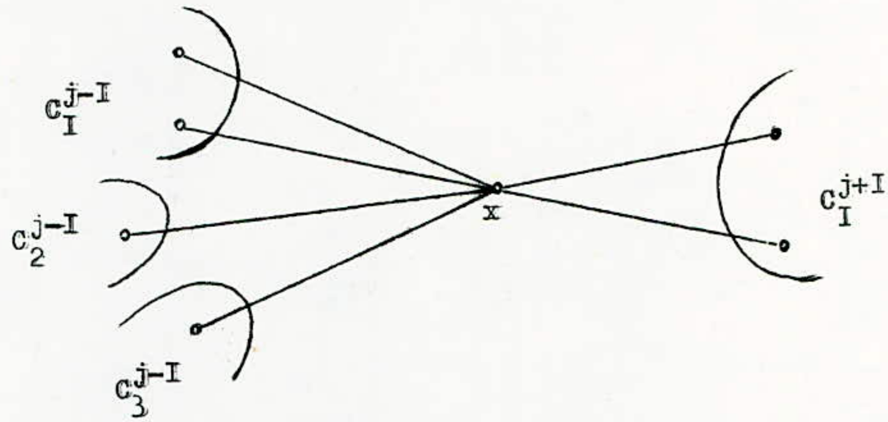
En considérant la relation précédente, associons aux sommets d'une couche le doublet (demi-degré interieur, demi-degré exterieur). Nous définissons ainsi une partition des sommets de la $j^{\text{ème}}$ couche en classes notées $(C_1^j, C_2^j, \ldots)$.

Si cette partition se réduità une seule classe pour toutes les couches considérées, l'algorithme est terminé et les H_s-classes maximales trouvées (cf. § suivant), la partition en couches étant identique à la partition en H_s- classes maximales.

3.I.2.2. Affinement de la partition associée à une couche

Associons à chaque sommet d'une couche 2 monômes , le monôme prédécesseur (successeur) sera constitué par la concaténation des prédécesseurs (successeurs), chaque élément étant représenté par la classe à laquelle il appartient (dernieres partitions des couches voisines.

Par exemple :



A x sont associés les 2 monômes : $\{(C_I C_I C_2 C_3)^{j-I}, (C_I C_I)^{j+I}\}$

On définit ainsi une nouvelle partition des éléments de cette couche inférieur ou égale à la précédente par la relation d'équivalence : 2 éléments seront dans une même classe si les couples de monômes associés sont les mêmes.

Remarque: aux éléments de la première couche, il suffira d'associer le monôme successeur.

3.I.2.3. Cas particuliers

Si aucun affinement défini précédemment n'est possible, l'algorithme est terminé. Les classes ainsi trouvées à l'intérieur d'une couche constituent des H_S- classes maximales

3.I.2.4. Remarques :

- L'affinement de la partition de la $j^{ème}$ couche peut amener le changement des monômes successeurs de la couche j-I, et des monômes prédécesseurs de la couche j+I, soit l'affinement des partitions associées aux couches j-I et j+I.

- les indices indiquant le numéro de couche C^k seront supprimés puisqu'aucune confusion ne sera possible.

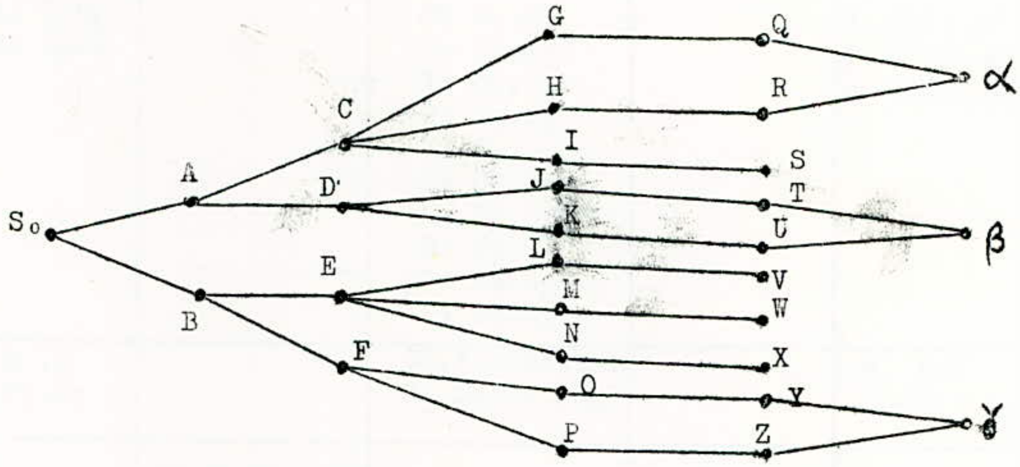
- il est facile de reconnaître, dans un premier temps, les éléments d'une couche appartenant à une même S - classe. Ces éléments ont même prédecesseur unique et même successeur, s'il existe.

Exemple :

Les monômes associés seront les mêmes pour ces sommets à chaque étape.

3.2. Exemples

3.2.1. Exemple I

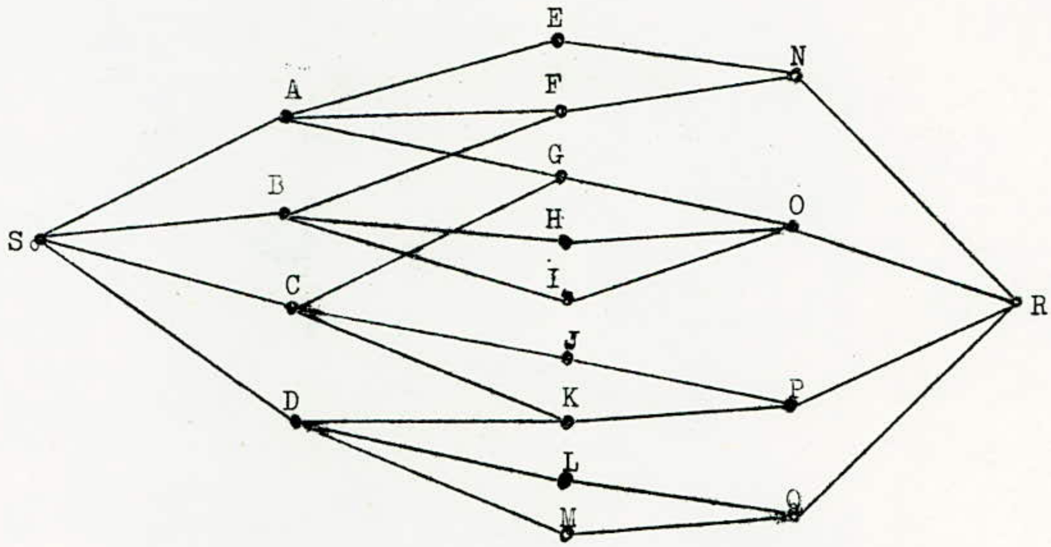


	Couche I	Couche 2	Couche 3	Couche 4	Couche 5
Partition initiale	A B C_I	C E , D F C_I , C_2	GHIJKLMNOP C_I	QRTUYZ , SVWX C_I , C_2	α β γ C_I
Monômes associés	A:C_IC_2 B:C_IC_2	C: $C_I,C_IC_IC_I$ D: C_I,C_IC_I E: $C_I,C_IC_IC_I$ F: C_I,C_IC_I	G: C_I,C_I H: C_I,C_I I: C_I,C_2 J: C_2,C_I K: C_2,C_I L: C_I,C_2 M: C_I,C_2 N: C_I,C_2 O: C_2,C_I P: C_2,C_I		
Nouvelle partition			GH, IIMN, JKOP C_I,C_2 , C_3		
Monômes associés		C:$C_I,C_IC_IC_2$ D:C_I,C_IC_I E:$C_I,C_2C_2C_2$ F:C_I,C_3C_3		Q: C_I,C_I R: C_I,C_I T: C_3,C_I U: C_3,C_I Y: C_3,C_I Z: C_3,C_I S: $C_2,\emptyset$ V: $C_2,\emptyset$ W: $C_2,\emptyset$ X: $C_2,\emptyset$	
Nouvelles partitions		C , E ,DF C_I, C_2,C_3		QR,TUYZ,SVWX C_I, C_2 , C_3	

	couche I	couche 2	couche 3	couche 4	couche 5
Monômes associés	A: $C_I C_3$ B: $C_2 C_3$		G: C_I, C_I H: C_I, C_I I: C_I, C_3 J: C_3, C_2 K: C_3, C_2 L: C_2, C_3 M: C_2, C_3 N: C_2, C_3 O: C_3, C_2 P: C_3, C_2		α: $C_I C_I, \emptyset$ β: $C_2 C_2, \emptyset$ γ: $C_2 C_2, \emptyset$
Nouvelles partitions	A ,B C_I, C_2		GH,I,JKOP,LMN C_I, C_2, C_3, C_4		α $\beta\gamma$ C_I, C_2
Monômes associés		C: $C_I, C_I C_I C_2$ E: $C_2, C_4 C_4 C_4$ D: $C_I, C_3 C_3$ F: $C_2, C_3 C_3$		Q: C_I, C_I R: C_I, C_I T: C_3, C_2 U: C_3, C_2 Y: C_3, C_2 Z: C_3, C_2 S: $C_2, \emptyset$ V: $C_4, \emptyset$ W: $C_4, \emptyset$ X: $C_4, \emptyset$	
Nouvelles partitions		C ,E ,D ,F C_I, C_2, C_3, C_4		QR,TUYZ,S,VWX C_I, C_2, C_3, C_4	
Monômes associés	stabilisation		G: C_I, C_I H: C_I, C_I I: C_I, C_3 J: C_3, C_2 K: C_3, C_2 O: C_4, C_2 P: C_4, C_2 L: C_2, C_4 M: C_2, C_4 N: C_2, C_4		α: $C_I C_I, \emptyset$ β: $C_2 C_2, \emptyset$ γ: $C_2 C_2, \emptyset$
Nouvelles partitions			GH,I,JK,OP,LMN C_I, C_2, C_3, C_4, C_5		α . $\beta\gamma$ C_I, C_2
Monômes associés				Q: C_I, C_I R: C_I, C_I T: C_3, C_2 U: C_3, C_2 Y: C_4, C_2 Z: C_4, C_2 S: $C_2, \emptyset$ V: $C_5, \emptyset$ W: $C_5, \emptyset$ X: $C_5, \emptyset$	

	couche I	couche 2	couche 3	couche 4	couche 5
Nouvelles partitions				QR,TU,YZ,S,VWX C_I, C_2, C_3, C_4, C_5	
Monômes associés			stabilisation		$\alpha : C_I, \emptyset$ $\beta : C_2, \emptyset$ $\gamma : C_3, \emptyset$
				stabilisation	α, β, γ
résultat	A,B	C,E,D,F	GH,I,JK,OP,LMN	QR,TU,YZ,S,VWX	α, β, γ

3.2.2. Exemple 2



	Couche I	Couche 2	Couche 3	Couche 4
Partition initiale	A B C D C_I	EHIJLM,FGK C_I , C_2	N P Q, O C_I , C_2	R C_I
Monomes associés	A: $C_IC_2C_2$ B: $C_IC_IC_2$ C: $C_IC_2C_2$ D: $C_IC_IC_2$			
Nouvelle partition	AC, BD C_I, C_2			
Monomes associés	E: C_I, C_I H: C_2, C_2 I: C_2, C_2 J: C_I, C_I L: C_2, C_I M: C_2, C_I F: C_I C_2 G: C_I C_I, C_2 K: C_I C_2,C_I			

	Couche I	Couche 2	Couche 3	Couche 4
Nouvelle partition		EJ, HI, LM, FK, G C_I, C_2, C_3 , C_4, C_5		
Monomes associés	A: $C_I C_4 C_5$ B: $C_2 C_2 C_4$ C: $C_I C_4 C_5$ D: $C_3 C_3 C_4$		N: $C_I C_4$, C_I P: $C_I C_4$, C_I Q: $C_3 C_3$, C_I O: $C_2 C_2 C_5$, C_I	
Nouvelles partitions	AC, B, D C_I, C_2, C_3		NP, Q, O C_I, C_2, C_3	
Monomes associés		E: C_I, C_I H: C_2, C_3 I: C_2, C_3 J: C_I, C_I L: C_3, C_2 M: C_3, C_2 F: $C_I C_2, C_I$ G: $C_I C_I$, C_3 K: $C_I C_3, C_I$		
Nouvelle partition		EJ, HI, LM, FK, G, K C_I, C_2, C_3, C_4, C_5, C_6		
Monomes associés	A: $C_I C_4 C_5$ B: $C_2 C_2 C_4$ C: $C_I C_5 C_6$ D: $C_3 C_3 C_6$		N: $C_I C_4, C_I$ P: $C_I C_6, C_I$ Q: $C_3 C_3, C_I$ O: $C_2 C_2 C_5, C_I$	
Nouvelles partitions	A, B, C, D C_I, C_2, C_3, C_4		N, P, Q, O C_I, C_2, C_3, C_4	

	Couche I	Couche 2	Couche 3	couche 4
Monômes associés		E : C_I,C_I H : C_2,C_4 I : C_2,C_4 J : C_3,C_2 L : C_4,C_3 M : C_4,C_3 F : C_IC_2,C_I G : $C_IC_3\ ,C_4$ K: $C_3C_4\ ,C_2$		
Résultats	A,B,C,D	E,J,IH,LM,F,G,H	N,P,Q,O	R

3.3. Application :

Isomorphisme entre deux graphes

Supposons que les partitions finales se réduisent à des classes d'un seul élément ou constituent des H_s-classes (facilement reconnaissables), nous pouvons alors en déduire un isomorphisme entre les deux graphes.

Nous allons d'abord étudier le problème dans le cas simple de deux graphes connexes G et G', non orientés, bipartites.

La généralisation aux graphes quelconques s'en déduira facilement.

Ensuite deux des exercices traités au chapitre 2 par la méthode d' UNGER seront traités par la méthode proposée qui permettra d'aboutir dans les deux cas à un isomorphisme.

3.3.1. Choix de S et S'

La recherche d'un isomorphisme entre G et G' se fera en appliquant l'algorithme précédent aux 2 graphes ce qui suppose un choix préalable de S et S'. Une solution unique existera dans certaines applications (graphe d'automate comportant un état initial, graphe ayant un centre...).

Dans le cas général, il convient de rechercher un couple de sommets S et S' tel que :

- S et S' ont même degré, mêmes demi-degrés
- Les structures en couches associées ont :
  - . même nombre de couches
  - . même nombre d'éléments dans une couche
  - . même partition initiale des sommets d'une couche, en tenant compte de la valeur du doublet (demi-degré intérieur, demi-degré extérieur) associé à chaque classe; l'ensemble de ces doublets pour une couche donnée doit être le même pour les 2 graphes.

La recherche de la structure en couches associée à un sommet S étant rapide (programmée à l'annexe), ces essais seront rapides.

3.3.2. Isomorphismes entre les 2 graphes

L'algorithme précédent sera appliqué aux 2 graphes, les classes des partitions initiales relatives à une même couche étant numérotées de la même façon (suivant la valeur du doublet).

En appliquant cet algorithme, on respectera la régle suivante: soit une partition P s'affinant en une partition $P' < P$.

En numérotant les classes de P' on respectera :

- l'ordre des classes de P : seront d'abord considérés les sommets de la première classe de P etc...
- au cours de l'éclatement d'une telle classe, on respectera dans la numérotation, un ordre lexicographique des variables figurant dans l'ensemble (monôme prédecesseur, monôme successeur).

Exemple :

reconsidérons l'exemple 2 (3.2.2.)

Nous avions dans la deuxième couche :

EHIJLM,FGK

C_I ,C_2

Puis les monômes associés : E: C_I,C_I

H: C_2,C_2

I: C_2,C_2

J: C_I,C_I

L: C_2,C_I

M: C_2,C_I

F: C_IC_2,C_I

G: C_IC_I,C_2

K: C_IC_2,C_I

La remarque précédente nous donne les nouvelles classes dans l'ordre unique suivant :

EJ,LM,HI,G,FK,

C ,C ,C ,C ,C

En respectant cette numérotation, G et G' sont isomorphes si, pour toute couche considérée, l'ensemble des couples de monômes associés aux classes de la partition relative à cette couche est <u>le même</u> pour G et G'.

La correspondance biunivoque entre les H_S - classes maximales d'une couche sera donnée ent associant les classes ayant même couple de monômes associés.

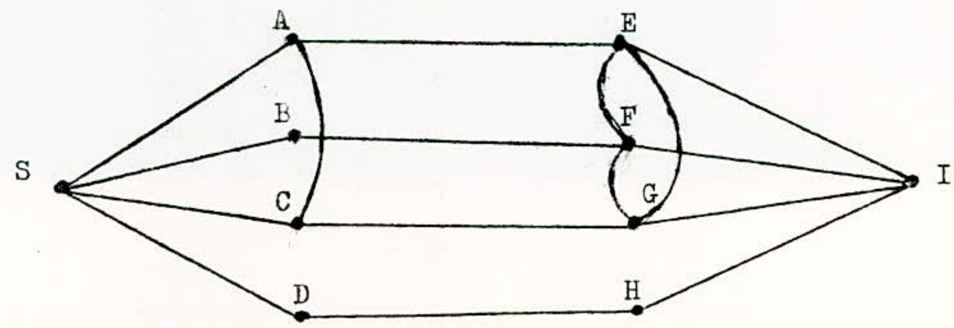
Par la remarque I.2.3.2. on peut en déduire l'ensemble des isomorphismes entre G et G' faisant correspondre S et S'.

3.3.3. Généralisation aux graphes quelconques

3.3.3.1. Graphes non bipartites

Une structure en couches, pour un sommet S donné , comportera des arêtes dont les extrêmités sont dans une même couche

Exemple :



L'algorithme précédent restera le même avec les deux remarques suivantes :

Remarque I
La partition initiale se fera selon le triplet (nombre de liaisons avec les éléments de la couche précédente, nombre de liaisons avec les éléments de la même couche, nombre de liaisons avec les éléments de la couche de rang supérieur)

Remarque 2
A chaque sommet sont maintenant associés 3 monômes : monôme prédécesseur, monôme traduisant les liaisons avec la même couche, monôme successeur; l'ordre lexicographique étant: ϕ ; C_I , C_2 ...

3.3.3.2. Application à l'exemple précédent :

	Couche 1	Couche 2	Couche 3
Partition initiale	A C , D C_1 C_2	E G , F , H C_1 C_2 C_3	I C_1
Monômes associés	A : C_1, C_2 B : , C_2 C : C_1, C_1 D : , C_3		
Nouvelle partition	B , D , AC C_1 , C_2 , C_3		

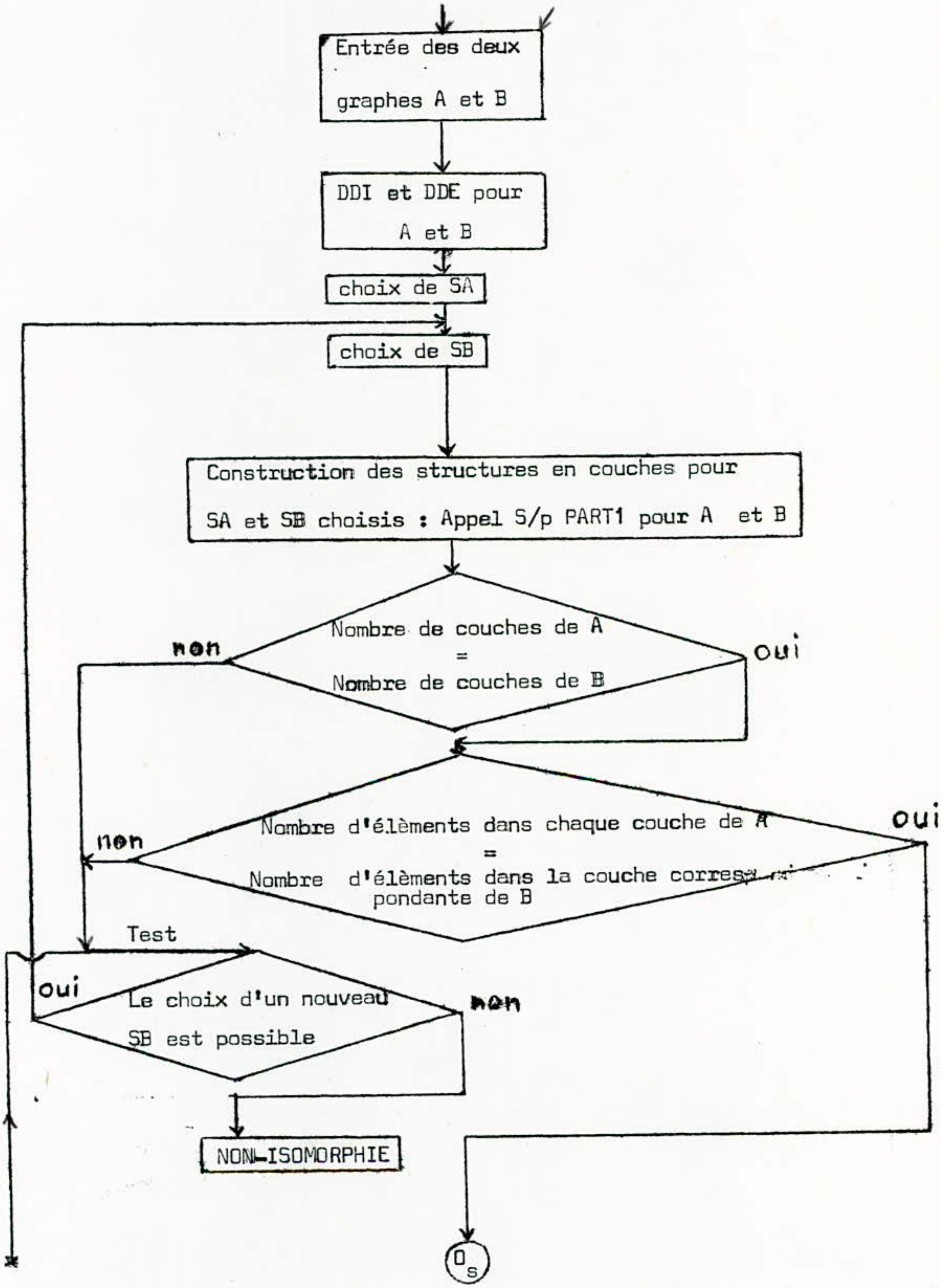
	Couche 1	Couche 2	Couche 3
Monomes associés		E : C_3, C_1 C_2, C_1 G : C_3, C_1 C_2, C_1 F ; C_1, C_1 C_2, H : C_2, $\emptyset$, C_1	
Solution	B,D, AC	EG,F,H	I

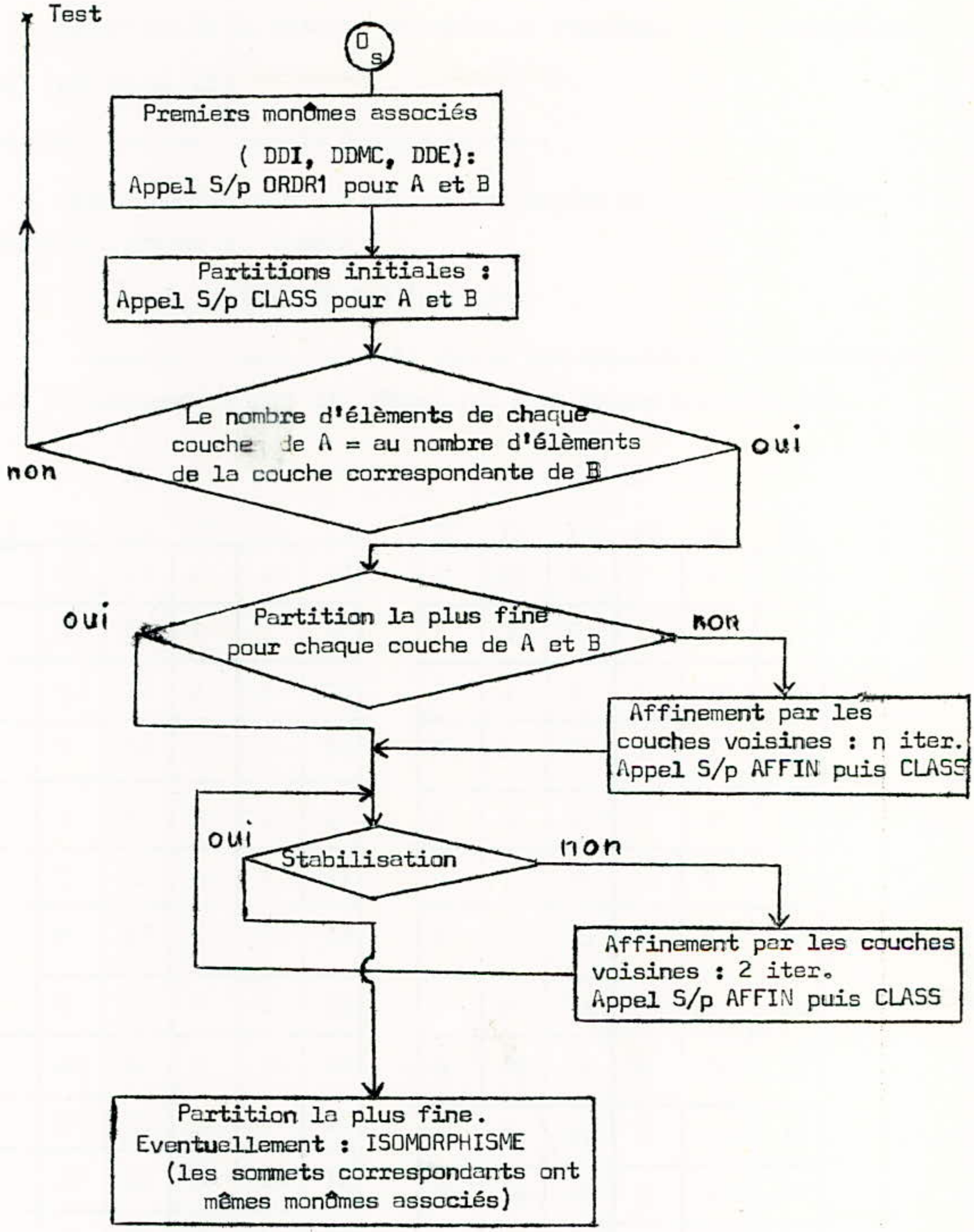
3.3.3.3. Graphes orientés

Nous disposons d'informations supplementaires; ceci se traduira dans l'algorithme précédent par un dédoublemeht des monômes associés, suivant le sens des arêtes.

En pratique cela correspondra à un encombrement mais l'algorithme sera plus rapide.

3.4. ORGANIGRAMME





Note : DDMC : Demi-degré pour les élèments de la même couche
n : nombre de couches pour chaque graphe.

Application de l'algorithme

(2) Construction de la structure en couches

a. Choix des sommets SA et SB

Les demi-degrés étant tous égaux (=3)et il n'y a pas de boucle, donc le choix de SA et SB est indifférent.

On prend SA =I , SB =I

b. Construction des couches correspondantes

La première couche de A est constituée par les sommets de A qui ont SA pour extrémité

la deuxième couche se compose des ommets de A , autres que SA dont une des extremités est un élément de la première couche.

On vérifie à chaque pas que les éléments d'une couche ne figurent pas dans les couches précédentes.

Le même processus est répété pour la troixième couche; on obtient:
taille des couches: nca (I) =3
nca (2) =6
nca (3) =4

Le même travail est fait pour le graphe B : construction de la structure en couches avec pour sommet origine: SB =I

On a : nombre de couches: ncb =4
ncb $\neq$ nca $\Rightarrow$ on prend un nouveau sommet origine : SB =2 , on a de même:
ncb =4 , pour SB=3,4,5
pour SB =6 , ncb =nca = 3

A ce stade nous avons obtenu les partitions en couches pour les 2 graphes. Nous pouvons, sur la base de ces partitions, entamer le processus d'affinement qui nous mènera éventuellement vers une isomorphie.

graphe A

CA

	II	I2	I3	2I	22	23	24	25	26	3I	32	33	34
I	2	I3	I4	3	5	7	I2	4	8	I0	6	9	II

$SA=I \text{ , } \left(ca_{k,kc} \quad \begin{array}{l} k= I,3 \\ kc=I,nca_k \end{array} \text{ ou } \begin{array}{l} nca_I=3 \\ nca_2=6 \\ nca_3=4 \end{array} \right)$

Graphe B

cb

	II	I2	I3	2I	22	23	24	25	26	3I	32	33	4I
I	2	I3	I4	3	5	7	I2	4	8	II	6	9	IO

	II	I2	I3	2I	22	23	24	25	3I	32	33	34	4I
2	I	3	5	I3	I4	4	II	6	7	I2	8	IO	9

3	2	4	II	I	5	I4	IO	I2	

4	3	5	I4	2	II	6	I	8	

5	2	4	6	I	3	I4	7	IO	

6	5	7	IO	2	4	8	I3	9	II	I	3	I4	I2

3 Prenons comme exemple le graphe A , nous construisons un tableau de partition qui donnera les partitions initiales.

Couche I, graphe A

sommet 2 : 6 composantes : 3x2, les 2 premières indiquent les liaisons avec SA comme il existe un arc non orienté allant de SA (=I) à 2 , ce qui équivaut à 2 arcs de sens contraires , les 2 premières composantes seront : $\binom{I}{I}$ = (DDI = I, DDE = I) les 2 composantes suivantes indiquent les liaisons avec les éléments de la même couche; comme de telles liaisons n'éxistent pas pour le sommet 2 les 2 composantes suivantes seront $\binom{0}{0}$ et enfin les 2 dernières qui expriment les liaisons avec les éléments de la couche suivante seront $\binom{2}{2}$, ç.à.d

2 arcs non orientés reliant 2 à 3 et 5

Pour la 2 ^{eme} couche on aura :
sommet de la 2eme couche (2 composantes expriment les liaisons avec les sommets de la couche précédente, 2 avec ceux de la même couche et les 2 dernières avec ceux de la couche suivante)

Pour la 3eme couche les deux dernières composantes seront nulles car c'est la dernière couche.

Un travail analogue est fait pour le graphe B (SB =6).

A l'aide des tableaux obtenus, nous obtenons de nouvelles partitions :

Graphe A

cla

II	I2	I3	2I	22	23	24	25	26	3I	32	33	34
I	I	I	I	I	I	I	I	I	I	2	2	I
I	I	I	I	I	I	I	I	I	I	2	2	I
0	0	0	I	I	I	0	2	I	2	I	I	2
0	0	0	I	I	I	0	2	I	2	I	I	2
2	2	2	I	I	I	2	0	I	0	0	0	0
2	2	2	I	I	I	2	0	I	0	0	0	0

$$\left(cla_{k_2,\ k,\ kc} \quad \begin{array}{l} k_2 = I,6 \\ k = I,3 \\ kc = I, nca_k \end{array} \right)$$

Graphe B

clb

II	I2	I3	2I	22	23	24	25	26	3I	32	33	34
I	I	I	I	I	I	I	I	I	2	3	2	3
I	I	I	I	I	I	I	I	I	2	3	2	3
0	0	0	0	0	I	0	I	0	I	0	I	0
0	0	0	0	0	I	0	I	0	I	0	I	0
2	2	2	2	2	I	2	I	2	0	0	0	0
2	2	2	2	2	I	2	I	2	0	0	0	0

(4) Les sommets ayant même composantes appartiennent à une même classe.

Ainsi les sommets 2,I3,I4 de la première couche constituent la classe C_I

La numérotation des classes se faisant dans l'ordre lexicographique, dans la 2eme couche nous avons : $C_I = \{I2\}$

$$C_2 = \{3,5,7,8\}$$
$$C_3 = \{4\}$$

Le nombre d'éléments dans chaque classe doit être le même pour les couches de même rang des deux graphes.

Ce qui n'est pas le cas; donc la structure en couche construite pour le graphe B (avec SB = 6) ne convient pas

Graphe A cma

II	I2	I3	2I	22	23	24	25	26	3I	32	33	34
I	I	I	2	2	2	I	3	2	I	2	2	I

$$\left(cma_{k,kc} \begin{array}{l} k = I,3 \\ kc = I,nca_k \end{array} \right)$$

Graphe B cmb

II	I2	I3	2I	22	23	24	25	26	3I	32	33	34
I	I	I	I	I	2	I	2	I	I	2	I	2

5 Nouvel essai pour SB = 7 : nouvelle structure en couches ncb = 4 ≠ nca , on change SB : SB = 8 ncb = 3

On construit le tableau de partition correspondant qui nous donne la nouvelle partition initiale des sommets de B .

On constate que les couches correspondantes des deux graphes ont même nombre d'éléments dans chaque classe.

On pourra donc commencer à affiner les dernières partitions de chaque couche par étude successive des couches voisines.

graphe B

	II	I2	I3	2I	22	23	24	25	26	3I	32	33	4I
7	6	8	I3	5	I0	9	I4	I	I2	2	4	II	3

	II	I2	I3	2I	22	23	24	25	26	3I	32	33	34
8	7	9	I4	6	I3	I0	I2	I	4	5	II	2	3

II	I2	I3	2I	22	23	24	25	26	3I	32	33	34
I	I	I	I	I	I	I	I	I	2	2	I	I
I	I	I	I	I	I	I	I	I	2	2	I	I
0	0	0	I	2	I	I	I	0	I	I	2	2
0	0	0	I	2	I	I	I	0	I	I	2	2
2	2	2	I	0	I	I	I	2	0	0	0	0
2	2	2	I	0	I	I	I	2	0	0	0	0

II	I2	I3	2I	22	23	24	25	26	3I	32	33	34
I	I	I	2	3	2	2	2	I	2	2	I	I

6 Il y a stabilisation pour une couche dès que la partition la plus fine est obtenue , ç.à.d. soit une partition où chaque classe est réduite à un seul élément soit une partition stationnaire, telle qu'une nouvelle tentative d'affinement ne l'améliore pas.

Nous allons donc construire un tableau dans lequel figureront les monômes associés à chaque sommet et les partitions qui s'en déduisent.
exemple :

couche I , sommet 2 :

$2 \in C_I$. Prédecesseurs de 2 : 2 (I,3,5)

. Successeurs de 2 : 2 (I,3,5)

comme c'ést la couche I le I^{er} monôme associé est 0-($\emptyset$); comme le sommet 2 n'a ni prédecesseur ni successeur dans la couche I , le 2eme monôme sera nul , le sommet I(=SA) éta nt exclu puisque ne faisant pas partie des couches; restent les sommets 3 et 5 qui appartient à C_2 (dans la couche 2), donc le 3eme monôme sera égal à 2 .

Comme pour les partitions initiales, la nouvelle partition des sommets de la couche I sera réalisée à partir du tableau précédent et en respectant l'ordre lexicographique.

Ainsi la nouvelle partition pour la couche I sera :

$C_I = \{I3\}$, $C_2 = \{2\}$, $C_3 = \{I4\}$

il y a stabilisation.

Dans la suite du calcul cette nouvelle partition remplace la partition initiale.

On prend maintenant la seule couche voisine : la 2eme , pour le sommet 3 par exemple les monômes associés seront 2 , 3 et I car 3(2,4,IO) , $2 \in C_2$, $4 \in C_3$, $IO \in C_I$.

Après la deuxième couche, on doit en principe considérer les 2 couches voisines: I et 3 mais comme la I^{ere} couche est stabilisée (partition la plus fine) on ne considére que la 3 ème couche

Noter que le 3eme monôme de la dernière couche est toujours nul car les prédécesseurs ou successeurs des sommets de cette couche ne peuvent se trouver que dans la couche précédente ou dans cette dernière couche.

Les nouvelles partitions obtenues sont les plus fines.

On fait la même chose que précedemment pour le graphe B.

Un isomorphisme éventuel exige que les monômes associés de chaque couche soi~~e~~ent les mêmes dans les 2 graphes

Nous constatons que ce n'est pas le cas pour la 2eme couche.

Nous devons par conséquent changer SB :

Graphe A

SA=I	2	I3	I4	3	5	7	I2	4	8	I0	6	9	I1	ca
	I	I	I	2	2	2	I	3	2	I	2	2	I	cma

2 I 3 3 4 2 I 6 5

			2	2	I	I	3	3	3	2	I	I
										4	5	
0	0	0	3	3	2	0	2	2	I	I	I	I
							2		2			2
2	I	2	I	2	2	I	0	2				
2	2	3				2						

parta

2	I	3	3	4	2	I	6	5	4	3	2	I	cma

$$\left(\text{parta}_{k_2,\ k,kc} \quad \begin{matrix} k_2 = I,9 \\ k = I,3 \\ kc = nca_k \end{matrix} \right)$$

Graphe B

SB=8

7	9	I4	6	I3	I0	I2	I	4	5	II	2	3
I	I	I	2	3	2	2	2	I	2	2	I	I
3	2	I	5	6	3	4	2	I				

			3	3	2	2	I	I				
0	0	0	2	2	2	3	3	0				
				2								
2	2	I	2	0	2	2	I	I				
3	2	2						I				

partb ↑

3	2	I	5	6	3	4	2	I					cmb

7 SB = 9 , nouvelle structure en couches.

nous constatons que pour cette valeur de SB comme pour SB = I0, II et I2 on a :

ncb $\neq$ nca

Pour SB = I3 nous construisons la structure en couches, le tableau de partition et le tableau d'affinement de la partition initiale.

Les monômes associés au sommet de la I^{ere} couche n'étant pas les mêmes que leurs correspondants pour le graphe A, donc un nouveau choix de SB est fait

Graphe B

	II	I2	I3	2I	22	23	24	25	3I	32	34	35	4I
9	8	IO	I2	7	I4	6	II	I3	[illegible]	4	5	3	2
IO	6	9	II	5	7	8	I2	3					
II	3	IO	I2	2	4	6	9	I3					
I2	9	II	I3	8	IO	3	I	7					
I3	I	7	I2	2	I4	6	8	9	II	3	5	4	IO

I	I	I	I	I	I	I	I	I	2	2	I	3
I	I	I	I	I	I	I	I	I	2	2	I	3
0	0	0	0	I	0	2	I	0	I	I	2	0
0	0	0	0	I	0	2	I	0	I	I	2	0
2	2	2	2	I	2	0	I	2				
2	2	2	2	I	2	0	I	2				

I	I	I	I	2	I	3	2	I	2	2	I	3

I3	I	7	I2	2	I4	6	8	9	II	3	5	4	IO
	I	I	I	I	2	I	3	2	I	2	2	I	3

0	0	0	
I	I	I	
2	3	2	
I	2	I	

8 SB = I4

le même cheminement que pour le sommet I3 est suivi , mais cette fois-ci les monômes associés sont les mêmes que ceux obtenus pour le graphe A (SA = I).

On en déduit l'isomorphisme cherché.

I4	I	4	8	2	I3	3	5	7	9	I2	II	6	IO
	I	I	I	I	I	I	I	I	I	2	I	2	I
	I	I	I	I	I	I	I	I	I	2	I	2	I
	0	0	0	2	I	I	I	I	0	I	2	I	2
	0	0	0	2	I	I	I	I	0	I	2	I	2
	2	2	2	0	I	I	I	I	2				
	2	2	2	0	I	I	I	I	2				

I	I	I	3	2	2	2	2	I	2	I	2	I

I4

I	4	8	2	I3	3	5	7	9	I2	II	6	I0
I	I	I	3	2	2	2	2	I	2	I	2	I
3	2	I	6	5	3	4	2	I				

			3	3	2	2	I	I	I	3	2	I
									5		4	
0	0	0	2	2	3	3	2	0	I	I	I	I
			2							2		2
2	2	I	0	2	I	2	2	I				
3	2	2						2				

3	2	I	6	5	3	4	2	I	2	4	3	I

ISOMORPHISME :

A	I	2	3	4	5	6	7	8	9	I0	II	I2	I3	I4
B	I4	4	3	2	5	6	2	I3	I2	II	I0	9	8	I

$$\left(iso_{i,j} \; \begin{matrix} i = I,I4 \\ j = I,2 \end{matrix} \right)$$

4.2. Exercice I (cf. Chap. II)

	II	I2	I3	2I	22	23
I	2	3	0	2	6	0
2	I	4	5	I	4	0
3	6	0	0	I	4	0
4	2	3	0	2	5	0
5	4	0	0	2	6	0
6	I	5	6	3	6	0

ca

	II	I2	I3	2I	22	23
I	2	3	6	2	6	0
2	I	4	0	I	5	0
3	6	0	0	I	5	0
4	5	0	0	2	6	0
5	2	3	5	4	5	0
6	I	4	0	I	3	0

cb

	II	I2	I3	2I	22
6	I	3	5	2	4

cla

	II	I2	I3	2I	22
5	2	3	4	I	6

clb

	II	I2	I3	2I	22
I	0	I	0	2	I
2	I	0	I	I	I
3	I	0	0	I	I
4	0	I	0	I	I
5	.I	0	I	0	I
6	I	I	I	0	0

cma

	II	I2	I3	2I	22
I	0	0	I	2	I
2	I	I	0	I	I
3	I	0	0	I	I
4	0	0	I	I	I
5	I	I	0	0	0
6	I	I	I	0	0

cmb

II	I2	I3	2I	22
2	3	I	2	I

II	I2	I3	2I	22
2	I	3	2	I

ISOMORPHISME :

A	I	2	3	4	5	6
B	2	I	4	6	3	5

Remarque : dans ce cas la partition initiale est la plus fine d'où immédiatement l'isomorphisme.

CONCLUSION

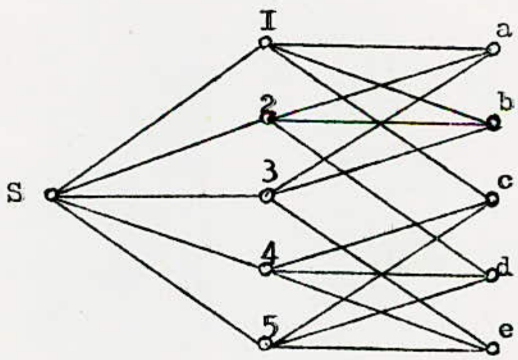
Les limites de la méthode proposée.

La méthode proposée n'est pas à proprement parler une méthode de recherche de l'isomorphie entre deux graphes comme la méthode d'UNGER par exemple , c'est une méthode puissante d'affinement des partitions. Dans beaucoup de cas , cette méthode permet d'obtenir la partition la plus fine et conduit à l'isomorphisme.

Mais cette méthode qui a le mérite d'utiliser relativement peu d'outils ne permet pas de conclure dans certains cas.

L'exemple suivant illustre ce fait :

Considérons le graphe A mis sous forme de couches :



Listes des successeurs et prédécesseurs

I(a,b,c)
2(a,b,d)
3(a,b,c)
4(c,d,e)
5(c,d,e)

---

a(I,2,3)
b(I,2,3)
c(I,4,5)
d(2,4,5)
e(3,4,5)

L'algorithme est inopérant :

Les demi-degrés sont tous égaux, la partition initiale est dans ce cas réduite à une classe , l'algorithme ne peut pas démarrer.

Une possibilité consisterait pas exemple à choisir des critères nouveaux pour essayer de répondre dans un tel cas;

on pourrait par exemple dire : 2 sommets appartiennent à une même classe s'ils ont même prédécesseur(successeur) (cf S-équivalence).

Mais la partition obtenue est stable (on ne peut plus l'affiner).

Une meilleure approche consisterait à étudier les automorphismes dans de tels cas.

Il ressort de cette tentative de trouver une méthode simple de recherche d'un isomorphisme qu'il est très difficile de ne pas faire appel à de nombreux outils tels que les fonctions de structure , et surtout à une procédure de choix car les possibilités d'affinement au moyen de régles strictes sont limitées.

D'ailleurs cette méthode pourrait être complétée par l'adjonction à l'algorithme donné au chapitre 3 d'une procédure de choix (une méthode d'hypothèses).

BIBLIOGRAPHIE

Claude BERGE : " Théorie des graphes et ses applications "
Collection universitaire de Mathématiques - DUNOD - I967.

J.P. STEEN : Méthode de recherche d'un isomorphisme entre deux graphes.
Math.Appli. Grenoble.

Stephen H. UNGER : " A Heuristic Program for Testing Pairs of Directed Line Graphs for Isomorphism "
D. TEICHROEW édit. Vol. 7 / Number 1 / January , I964.

F. HARARY, R. NORMAN, D. CARTWRIGHT : " Initiation à la Théorie des graphes orientés " .
DUNOD - Paris I968.