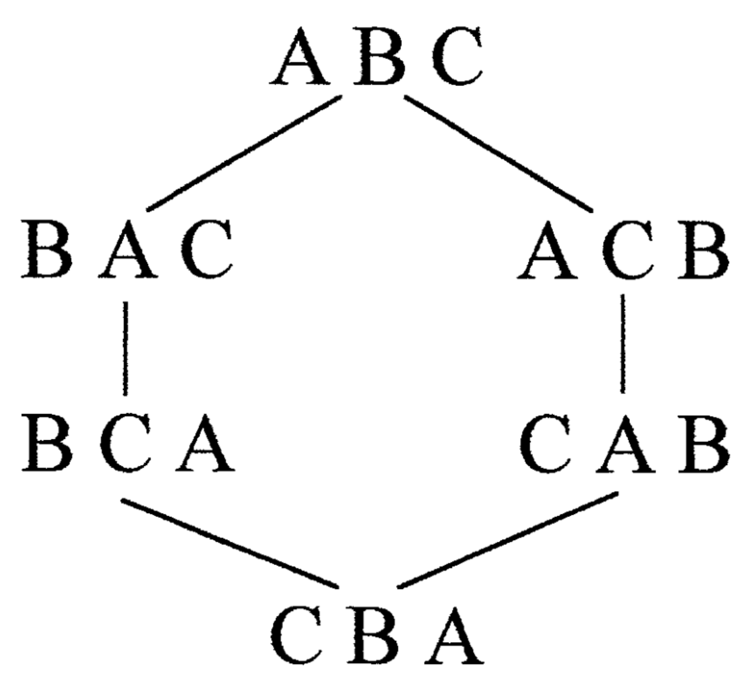
arrivés à ce stade, nous nous demandons s'il est nécessaire de recommencer cette procédure pour chacune des permutations de tous les mots formés de plus de trois lettres.

il y a 6 combinaisons de 3 lettres, 24 combinaisons de 4 lettres, *n!* combinaisons pour les mots de *n* lettres.

or il n'en est rien ! l'ensemble *COND* nous permet de construire une machine abstraite, un automate, qui va nous simplifier la tâche. reprenons les mots de 3 lettres. il y en a 6 :



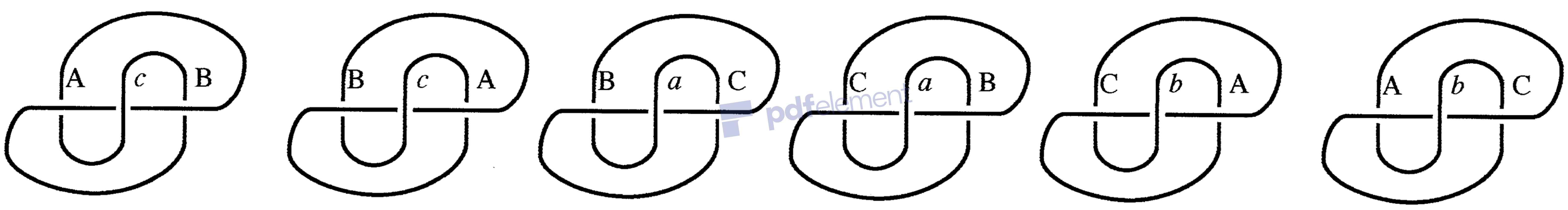
à chacun de ces 6 mots correspond un mot *abc* tel que, normalement, leur intrication donne un mot noué.

la procédure est toujours la même :

A B C	B A C	B C A	C B A	C A B	A C B
c a b	*c b a*	*a b c*	*a c b*	*b c a*	*b a c*

et les mots-lecture correspondants sont :

A*c*B*a*C*b* B*c*A*b*C*a* B*a*C*b*A*c* C*a*B*c*A*b* C*b*A*c*B*a* A*b*C*a*B*c*



ces 6 mots-lecture donnent le même nœud : il n'y a qu'un seul nœud à 1 rond et 3 croisements. mais 6 mots pour un seul état, c'est surabondant !

1ère **normalisation**

la *normalisation* permet de comparer, c'est-à-dire d'identifier, les permutations des mots-écritures apparemment différents. elle consiste à réécrire les permutations majuscules en un mot base écrit dans l'ordre naturel, alphabétique ABC., numérique 123., selon l'équivalence $(X_r, x) \mapsto (N_r, n)$, où $X_r \in$ mot donné et $N_r \in$ mot base, l'indice **r** représentant le rang de la lettre dans son mot. lorsque tous les éléments du mot donné ont été traduits en ceux du mot base, celui-ci devient le mot référent auquel pourront être comparées les autres permutations qui auront subi la même procédure de normalisation.

exemple : le mot **B***c***A***b***C***a*. les majuscules sont dans l'ordre **BAC**. pour les normaliser en **ABC**, voici comme l'on procède :

1°) on écrit le mot référent majuscule sous les majuscules du mot donné; les minuscules du mot référent sont encore inconnues à ce stade. on les représente par les minuscules *x*, *y* et *z*.

B*c***A***b***C***a*	mot donné
A*x***B***y***C***z*	mot référent

2°) le **B** donné est **A** référent, et *b* donné est *y* référent, donc par échange des valeurs *y* passe de *b* donné à *a* référent et le mot référent devient **A***x***B***a***C***z*. ensuite, le **A** donné est **B** référent et *a* = *z* devient par échange des valeurs *z* = *b*; le mot référent devient **A***x***B***a***C***b*, ce qui entraîne *x* = *c* . le mot référent est finalement **A***c***B***a***C***b*.

lectures
le mot donné **B**_c_**A**_b_**C**_a_ peut être circularisé en **A**_b_**C**_a_**B**_c_. appliquons la procédure de normalisation à ce mot :

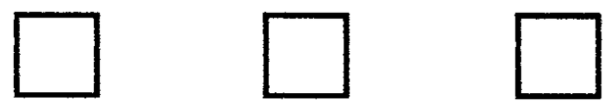
A_b_**C**_a_**B**_c_
A_x_**B**_y_**C**_z_

A est **A** et $y = a$, le mot référent devient **A**_x_**B**_a_**C**_z_ ; **C** donné est **B** référent et z passe de c donné à b référent, le mot référent devient **A**_x_**B**_a_**C**_b_ et finalement **A**_c_**B**_a_**C**_b_, soit le même mot référent que précédemment.
nous observons que si le mot donné se lit dans le sens de la flèche à partir d'une quelconque de ses lettres par circularité **B**_c_**A**_b_**C**_a_, le mot référent se lit, lui, en lecture inverse **A**_c_**B**_a_**C**_b_.

2ème *normalisation* ou **renormalisation**

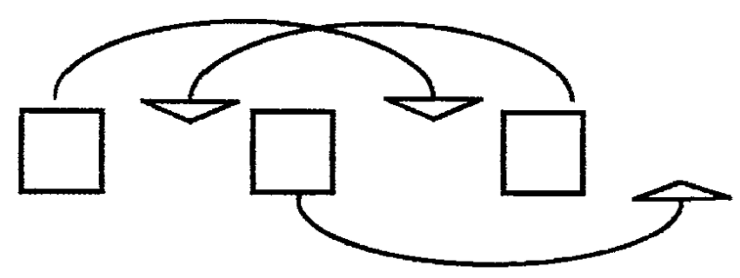
la machine *COND* va là aussi nous simplifier la tâche. en voici la construction.

1. nous dessinons autant de cases vides que de lettres majuscules :

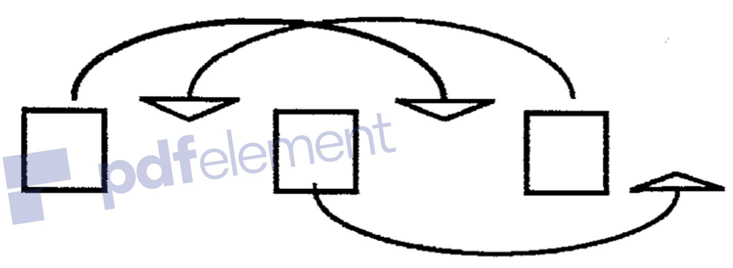


COND nous rattache à une théorie formelle des rangs.

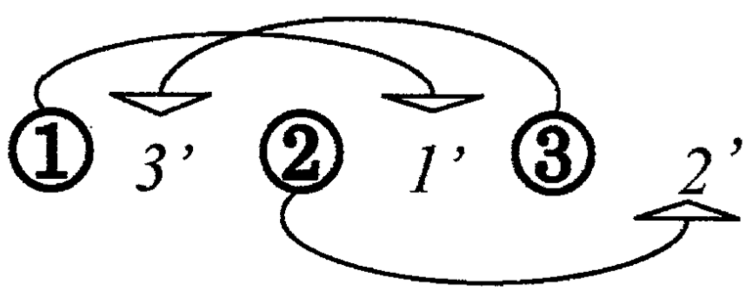
2. appliquons *COND* à chacune des cases :

$\mathscr{A}_3$: 

quelle que soit la lettre majuscule placée dans une case son double minuscule se place nécessairement à l'aboutissement de la flèche.
ainsi nous pouvons considérer

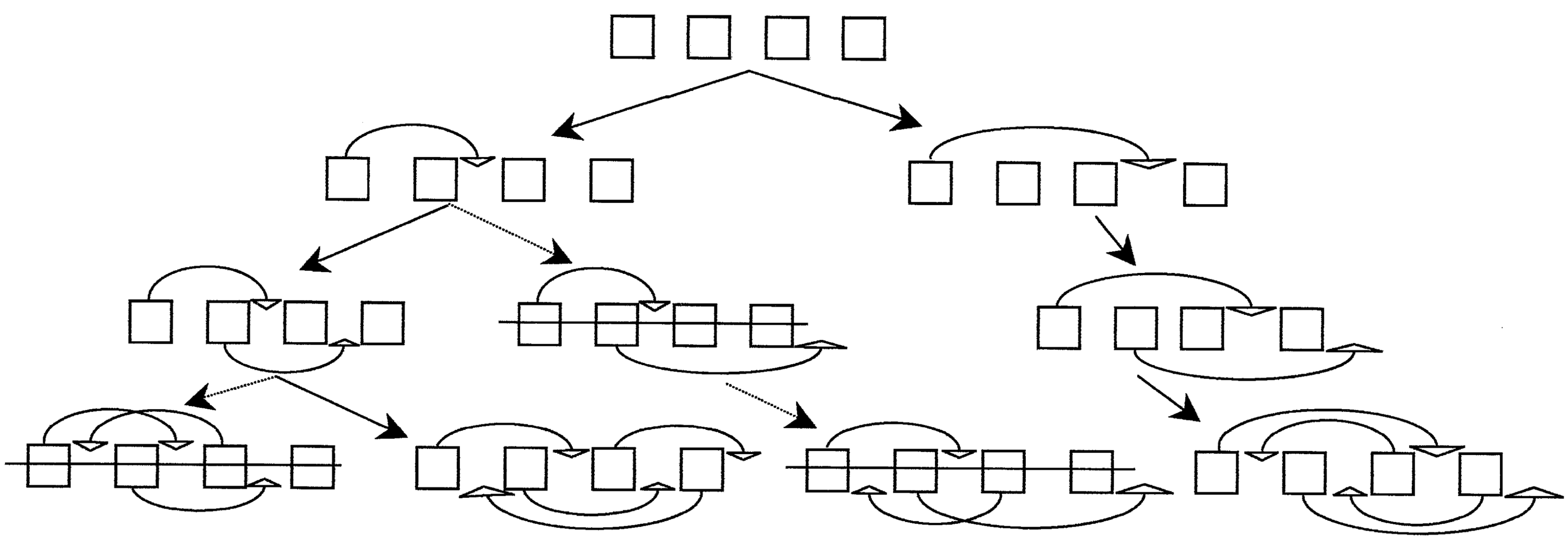
$\mathscr{A}_3$: 

comme le *3*-automate qui, à toute combinaison ternaire lui fait correspondre son mot-lecture noué. si maintenant, nous associons un nombre à chaque case et le même nombre primé à la place qu'occupera son compagnon dual, nous obtiendrons la machine informée $\mathscr{A}_3$:



c'est-à-dire **1**_3'_**2**_1'_**3**_2'_.

c'est le **nom**(*bre*) de l'état du nœud et du *3*-automate.
voyons ce qu'il en est du *4*-automate $\mathscr{A}_4$. explorons l'arbre des possibilités :
(les chemins en pointillé et les impasses auxquels ils aboutissent, sont barrés rétroactivement)

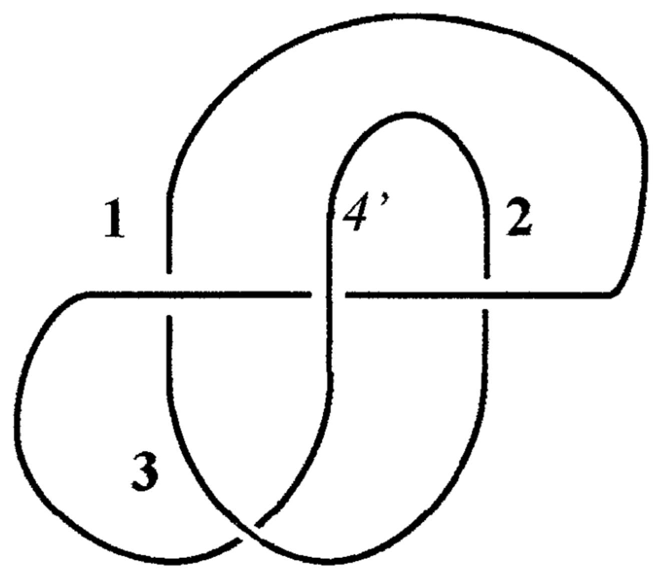


impossible $\mathscr{A}^1_4$: **1**_4'_**2**_1'_**3**_2'_**4**_3'_ *impossible* $\mathscr{A}^2_4$: **1**_3'_**2**_4'_**3**_1'_**4**_2'_

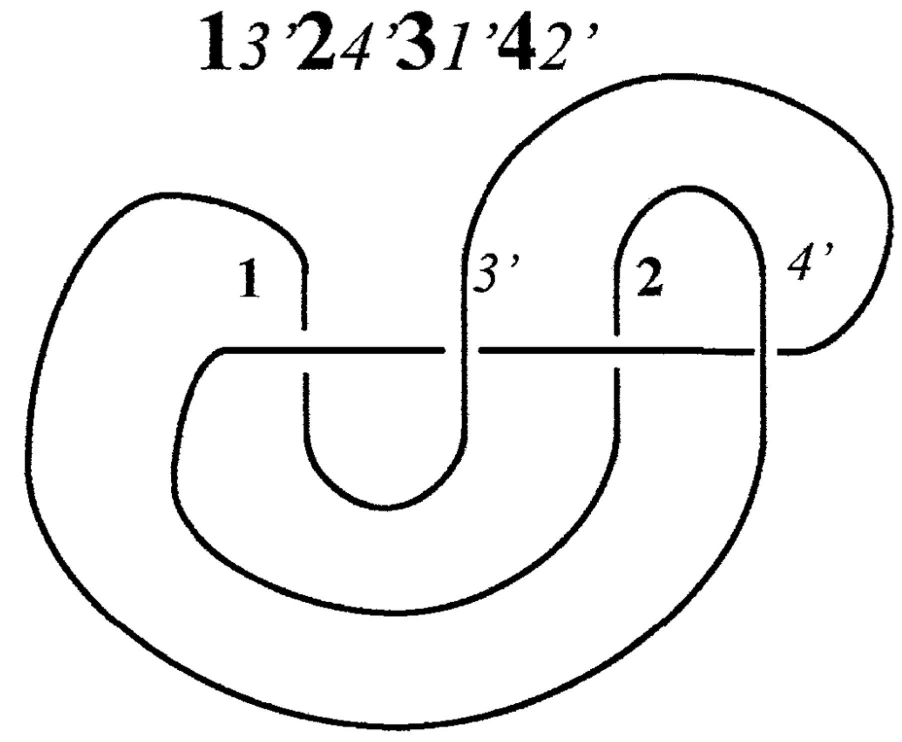
josef bayéma

il n’y a que 2 mots-lecture possibles :

1*4’***2***1’***3***2’***4***3’*



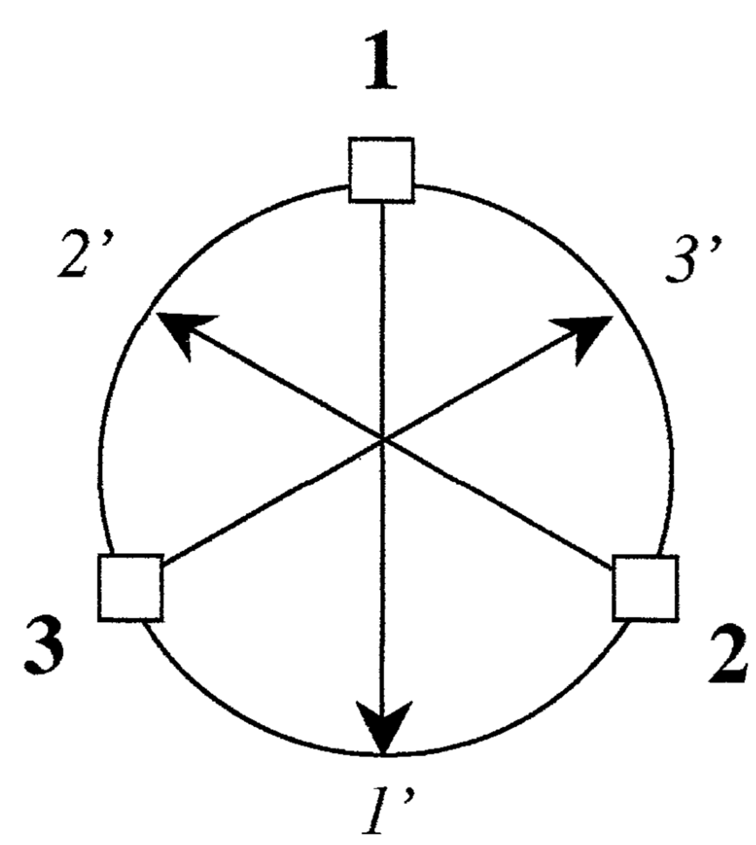
1*3’***2***4’***3***1’***4***2’*



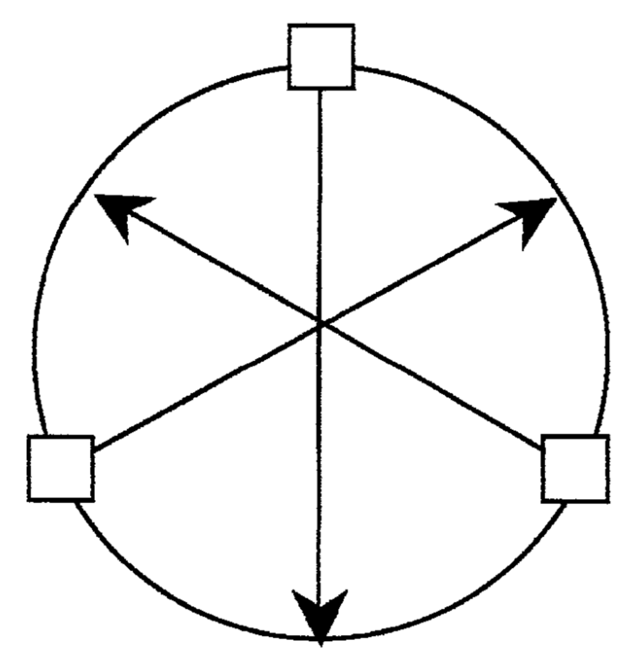
qui représentent le même nœud ; il n’y a qu’un seul nœud à 1 rond et à 4 croisements que, pour simplifier, nous écrivons (*4’ 1’ 2’ 3’*) et (*3’ 4’ 1’ 2’*). ces mots sont en translation circulaire d’incrément *1* l’un de l’autre.

au *n*-automate on peut faire correspondre le dessin de sa structure de la façon suivante, dite aussi espace d’écriture ; le cercle étant le support des bouclets.

3-automate $\mathscr{A}_3$ **1***3’***2***1’***3***2’*

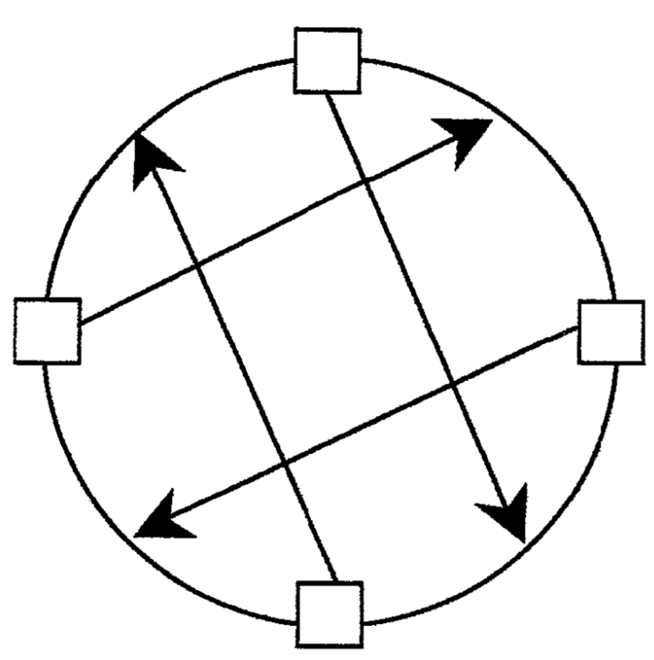


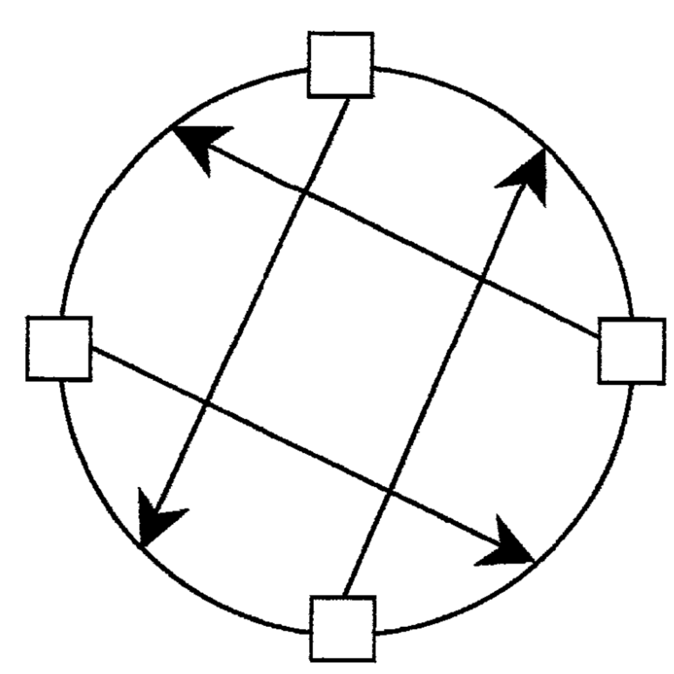
soit :



4-automate $\mathscr{A}^1_4$ **1***4’***2***1’***3***2’***4***3’*

4-automate $\mathscr{A}^2_4$ **1***3’***2***4’***3***1’***4***2’*



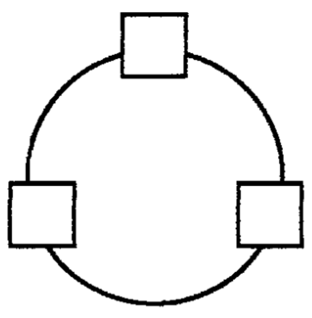
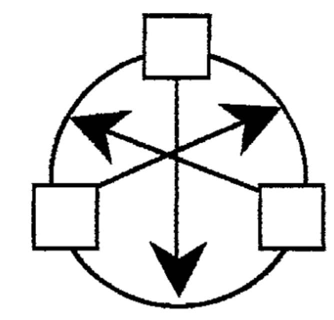


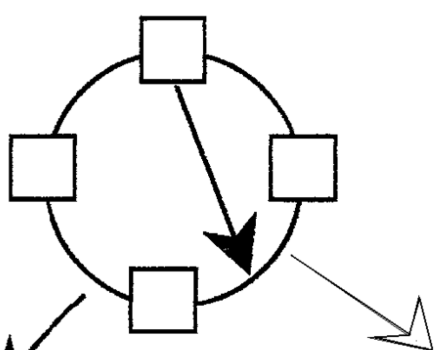
B. ***rattachement à la notion de bouclet dans l’état***

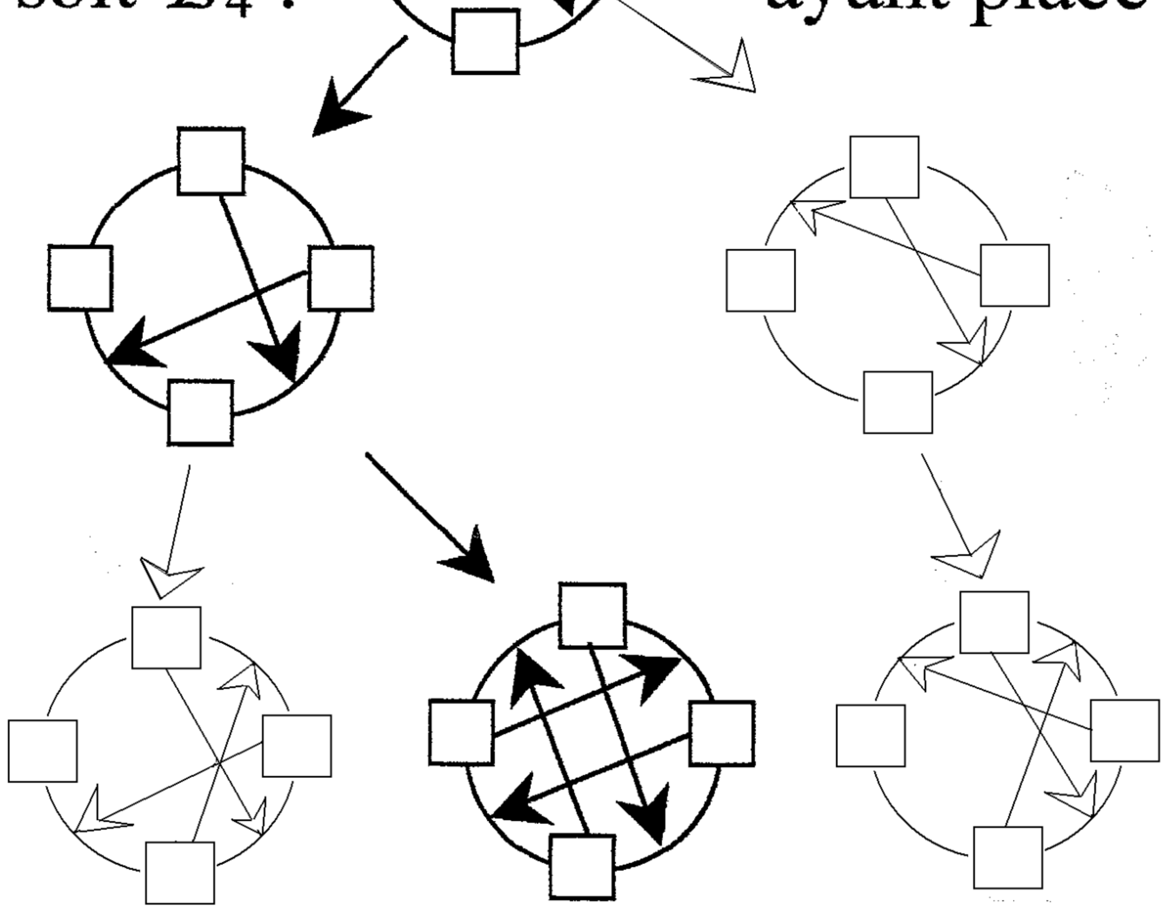
l’espace d’écriture des mots noués montre qu’il y a deux chemins, les bouclets, sur le cercle pour aller d’un carré à la pointe de sa flèche ; dans le cas du trèfle de quelque côté qu’on parcourre le bouclet, il faut compter 3 pas : un pas est parcouru lorsqu’on passe dans le mot-lecture d’un caractère à celui qui lui est immédiatement voisin, c’est-à-dire aussi sur le cercle lorsque l’on saute d’un carré à la pointe de flèche qui lui est immédiatement voisine et réciproquement. la lecture directe se faisant dans le sens des aiguilles d’une montre, la complémentation à *2.c* dans le sens trigonométrique.

un bouclet simple, **X***x’* ou *x’***X**, est le parcours nécessaire pour partir d’un croisement et aboutir au second brin dudit croisement. la somme des bouclets ou bouclet total est le chemin qu’il faut parcourir pour partir d'un brin d'un croisement et revenir à ce même brin dans la position de départ. le bouclet simple est dit 1/2 parcours, ce qui ne signifie pas que la quotité des deux bouclets simples soit identique. c'est cependant le cas pour le trèfle, pour lequel chaque bouclet vaut *3* et donc le bouclet complet égale 2×3, soit *6*. dans le cas du listing, un bouclet vaut *3* et l'autre vaut *5*, soit au total *8*. l'espace d'écriture du listing montre que ce qui vaut *3* dans un sens vaut *5* dans l'autre et les deux espaces d'écriture du listing sont inverses spéculaires l'un de l'autre.

les espaces d'écriture sont aussi un moyen de former les combinaisons qui donneront des mots noués en utilisant le plus simplement du monde les possibilités offertes pour envoyer les flèches à partir des carrés. un tel usage de l'espace d'écriture montre d'emblée pourquoi il ne peut y avoir de nœud à 1 rond et 2 croisements. nous noterons désormais un espace d'écriture par la lettre $\mathcal{E}$.

soit $\mathcal{E}_3$:  *COND* permet de placer immédiatement les flèches : 

soit $\mathcal{E}_4$:  ayant placé la première flèche – la symétrie permet de ne faire qu'un seul envoi,



deux cas sont à envisager pour la deuxième flèche.

de même deux cas sont à envisager pour la troisième flèche du $\mathcal{E}$ de gauche et une seule possibilité pour le $\mathcal{E}$ de droite. finalement, une seule possibilité s'affirme.

impossible **1***4'***2***1'***3***2'***4***3'* *impossible*

deux $\mathcal{E}$ strictement symétriques représentent le même état de nœud, le même nœud si celui-ci n'a qu'un état.

mais là aussi cette méthode risque de se montrer compliquée au fur et à mesure que l'on ajoute des croisements, c'est-à-dire des cases dans le cercle d'$\mathcal{E}$ et par conséquent des flèches.

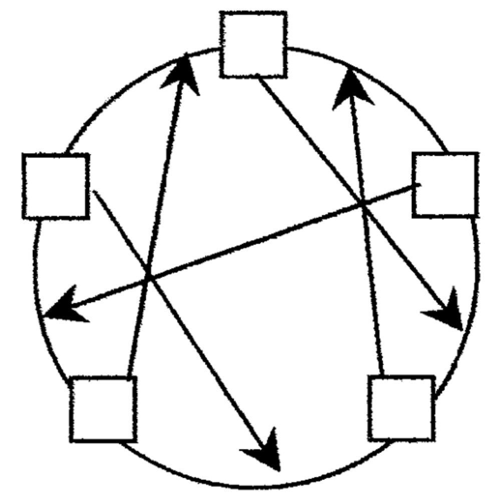
nous pouvons découvrir une autre méthode en utilisant la circularité des permutations dans le cadre de l'usage de *COND*.

- ***circularité et calcul des permutations***

nous allons effectuer cette méthode à partir des mots noués et de certaines observations. par exemple, tout mot commence par les cases **1** et **2** ; eh bien ! nous saurons ainsi qu'AUCUN mot ne commencera jamais par *1' 2'* – nous n'écrirons que le bout de la flèche puisque c'est lui qui est gouverné par *COND*. nous pourrons donc commencer tout mot par *3'* ! puis par *4'*, puis par *5'*, etc. nous gagnons ainsi DEUX places dans le calcul des permutations qui vont, rappelons-le, jusqu'à n, ce qui, AU DÉPART, nous fait passer de $n!$ à $(n-2)!$. de plus, par circularité, tous les mots obtenus à partir d'un mot souche ne donnent pas nécessairement un mot noué. ainsi, pour faire simple, si nous avons *3'1'2'*, nous savons que le mot obtenu est le seul ; si nous avons *3'4'1'2'*, le mot produit par circularité est *4'1'2'3'*, et nous savons qu'il est unique. ces deux mots se produisent donc l'un l'autre par circularité. à partir d'ici, puisque les MAJ se présentent toujours dans l'ordre naturel, nous simplifions l'écriture des mots lecture en n'écrivant que les parties primes.

- ***mots-nœuds***

prenons maintenant un mot à 5 places. un Æ nous donne

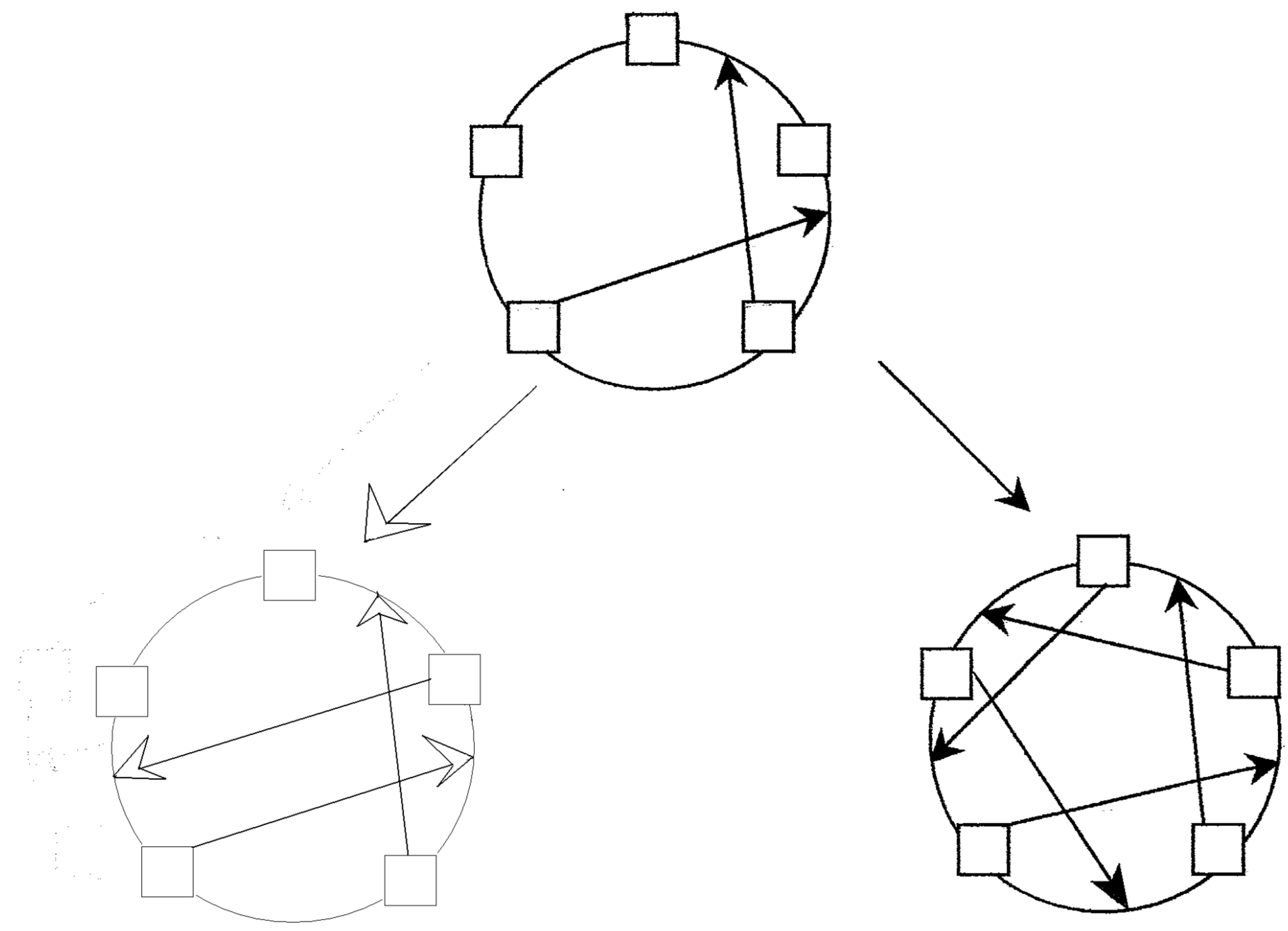


soit *3'1'5'2'4'*

il y a $5! = 120$ mots possibles. soit $5 \times 24 = 5 \times 4!$.

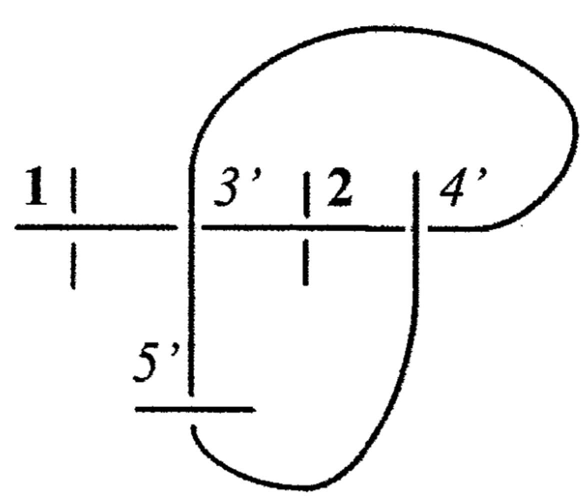
nous retirons *1'* et *2'* des deux premières places, ce qui donne $(5 - 2) \times 4! = 3 \times 4!$, c'est-à-dire $120 - 48 = 72$ combinaisons possibles.

circularisons *3'1'5'2'4'* : *1'5'2'4'3'* est exclu à cause de *1'* en première place (*COND2*), puis *5'2'4'3'1'*, exclu pour une même raison (n'oublions pas que chaque case inhibe 2 positions Ø □ Ø !), puis *2'4'3'1'5'* exclu, puis *4'3'1'5'2'* exclu, car *3'***3**. donc, à partir du mot *3'1'5'2'4'*, il n'est pas possible d'obtenir un autre mot noué. on peut donc prospecter *4'* comme second de *3'*. le mot est *3'4'5'1'2'*. il est assez facile de vérifier qu'il est le seul. il est naturel de penser à utiliser les différentes méthodes pour obtenir ces résultats. par exemple :



il s'agit bien de *3'4'5'1'2'*.

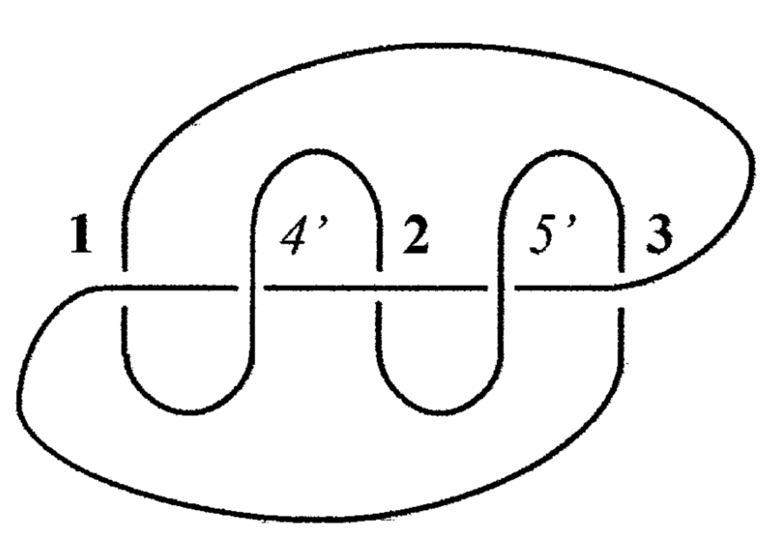
par circularité, nous obtenons *4'5'1'2'3'*, qui est bien un mot noué ; continuons : *5'1'2'3'4'* est aussi un mot noué. observons maintenant quels mots noués dessinent un nœud. essayons *3'4'5'1'2'* ; le mot complet est **1***3'***2***4'***3***5'***4***1'***5***2'*. dessinons ce mot. arrivés à **4**, nous ne pouvons atteindre son suivant *1'* sans créer un croisement supplémentaire.



nous voyons donc que ce mot noué n'est pas un mot-nœud. tous les mots noués ne sont donc pas nécessairement des mots-nœuds. les mots noués offrent l'avantage de restreindre l'ensemble de choix pour dessiner les nœuds.

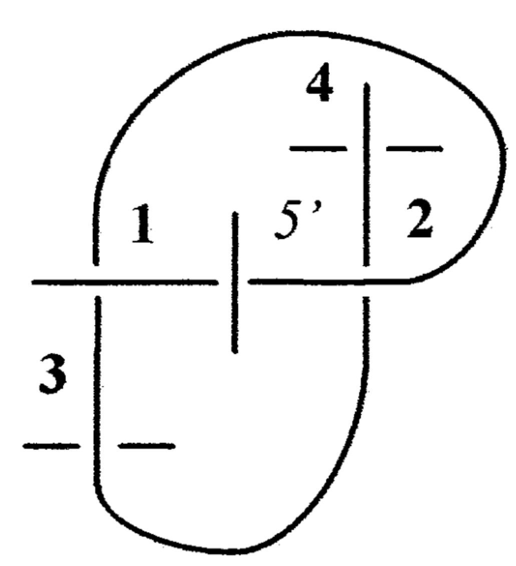
observons ce qui se passe avec les mots obtenus à partir de *3'4'5'1'2'* par circularité :

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



donc ce mot est un mot-nœud

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



donc ce mot n'est pas un mot-nœud

voici la liste de tous les mots noués à 5 lettres ; les mots barrés ne sont pas des mots-nœuds :

~~*3'1'5'2'4'*~~
~~*4'5'1'2'*~~ → *4'5'1'2'3'*, ~~*5'1'2'3'4'*~~
5'1'2'4'
~~*2'1'4'*~~

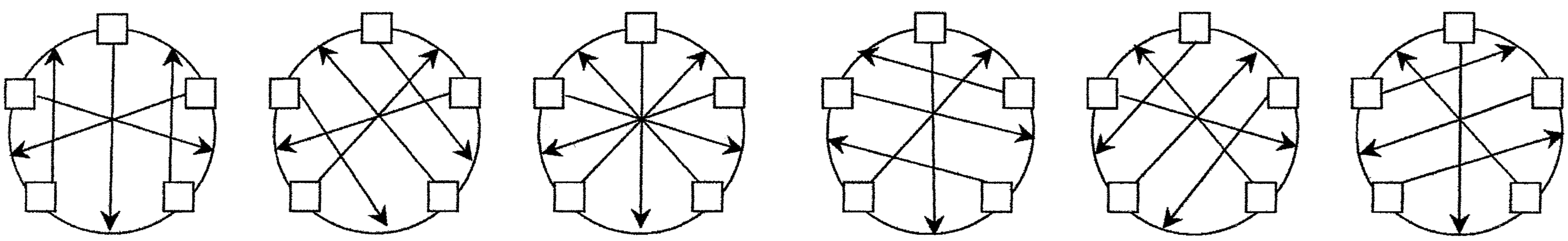
4'1'5'2'3'
~~*3'2'*~~
5'1'3'2'
2'1'3'

5'4'1'2'3'
~~*3'2'*~~
~~*2'1'3'*~~

il y a 13 mots noués parmi lesquels 6 mots-nœuds. il y a 120 permutations de 5 lettres. de ces 120 nous retirons les *1'* et *2'*, soit *120 - 48 = 72*. parmi ces 72, il y a 20 non noués commençant par *3'*, 19 commencent par *4'* et 20 commencent par *5'*, soit *20 + 19 + 20 = 59* et *72 - 59 = 13*.

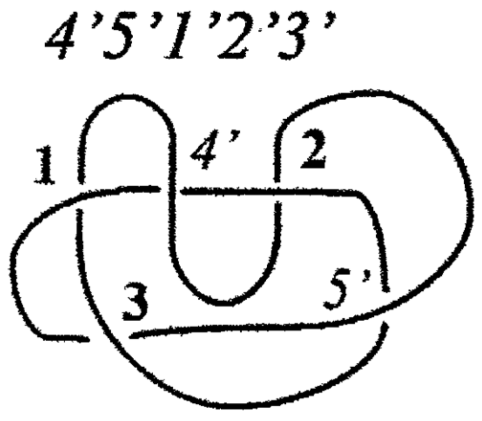
les Ɛ des 6 mots-nœuds sont les suivants :

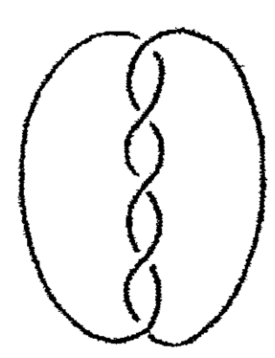
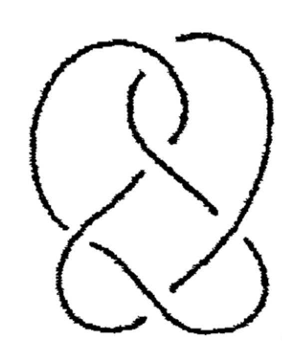
3'5'1'2'4' *4'1'5'2'3'* *4'5'1'2'3'* *4'5'1'3'2'* *4'5'2'1'3'* *5'4'1'2'3'*



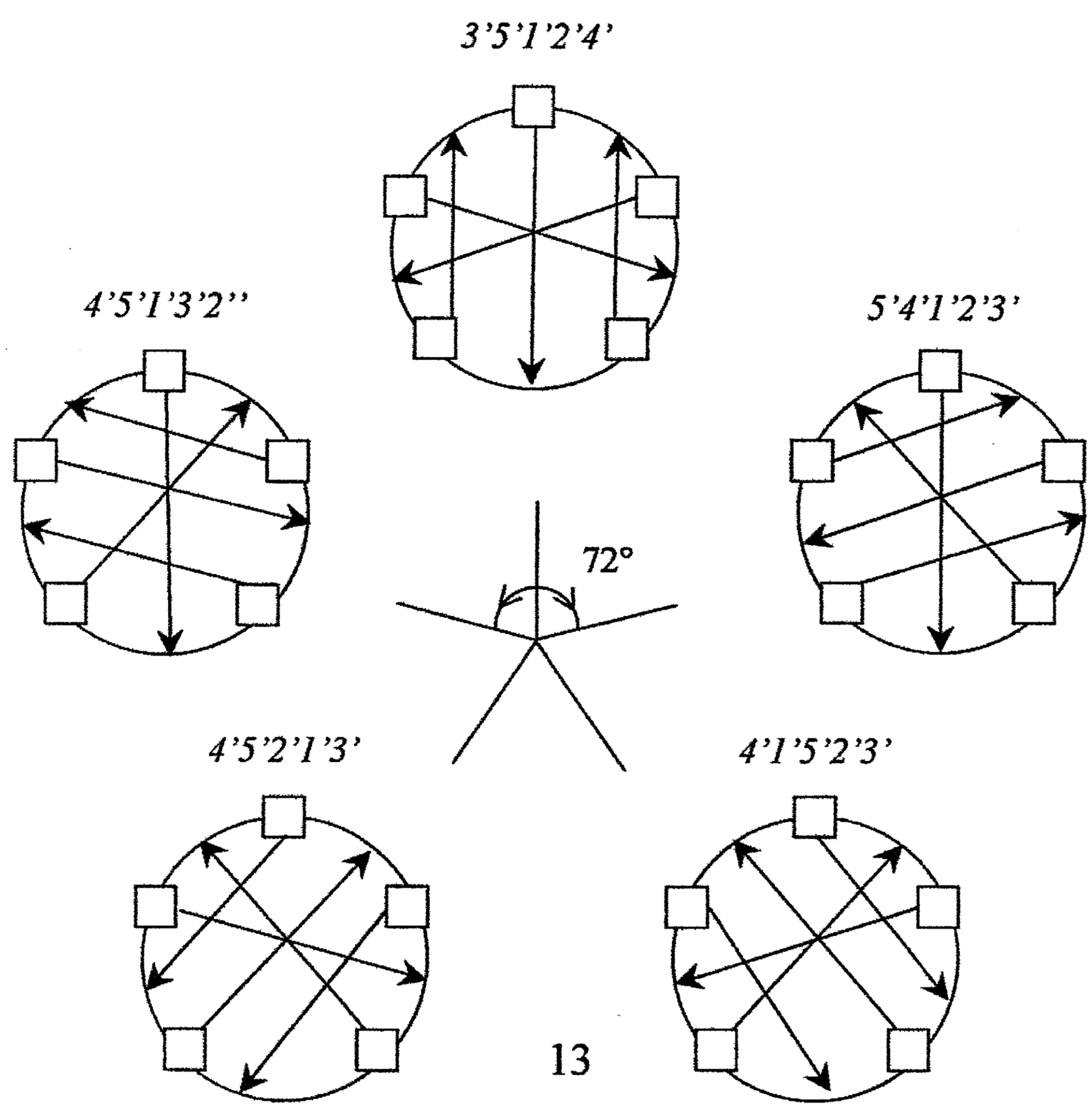
tous ces Ɛ ont en commun de posséder un point de rencontre des flèches central et ils sont les seuls parmi les mots noués. l'Ɛ de *4'5'1'2'3'* en étoile donne un nœud différent des autres Ɛ, à trois flèches parallèles dont une de sens différent, trois flèches se croisant au centre, qui donnent le même nœud. il n'y a donc que 2 nœuds à 1 rond et 5 croisements :

4'5'1'2'3'

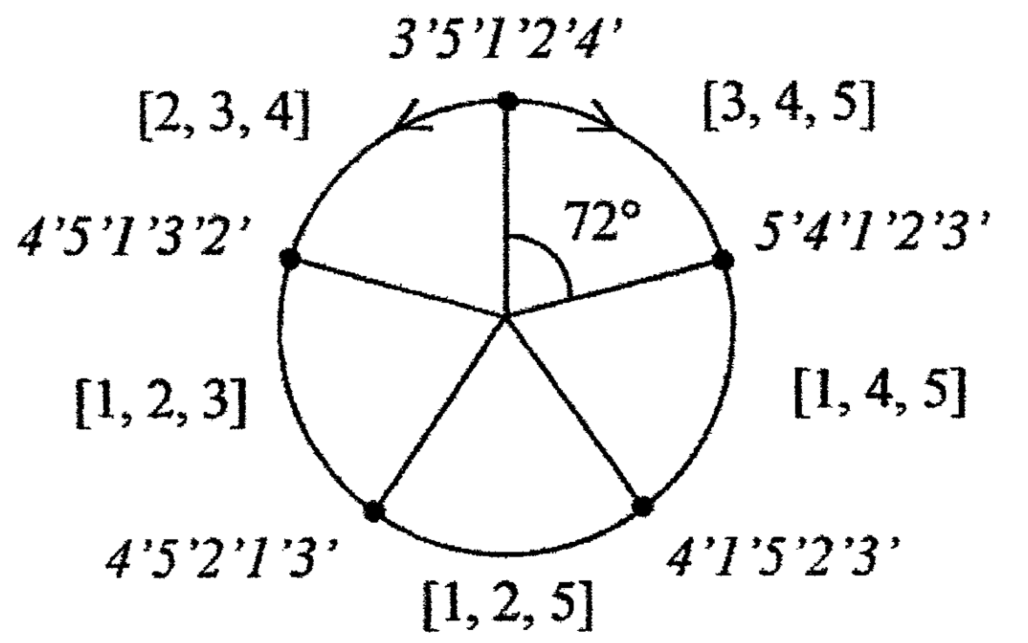


soit  et, pour les autres mots-nœuds, .

on remarque que les 5 mots-nœuds donnant le même nœud ont une morphologie identique différente de la morphologie de *4'5'1'2'3'*. ces 5 Ɛ forment un ensemble de 5 rotations de 2π/5.



pour observer la rotation, il suffit de suivre le mouvement des flèches parallèles, dans un sens ou dans l'autre. situation que l'on peut présenter ainsi :

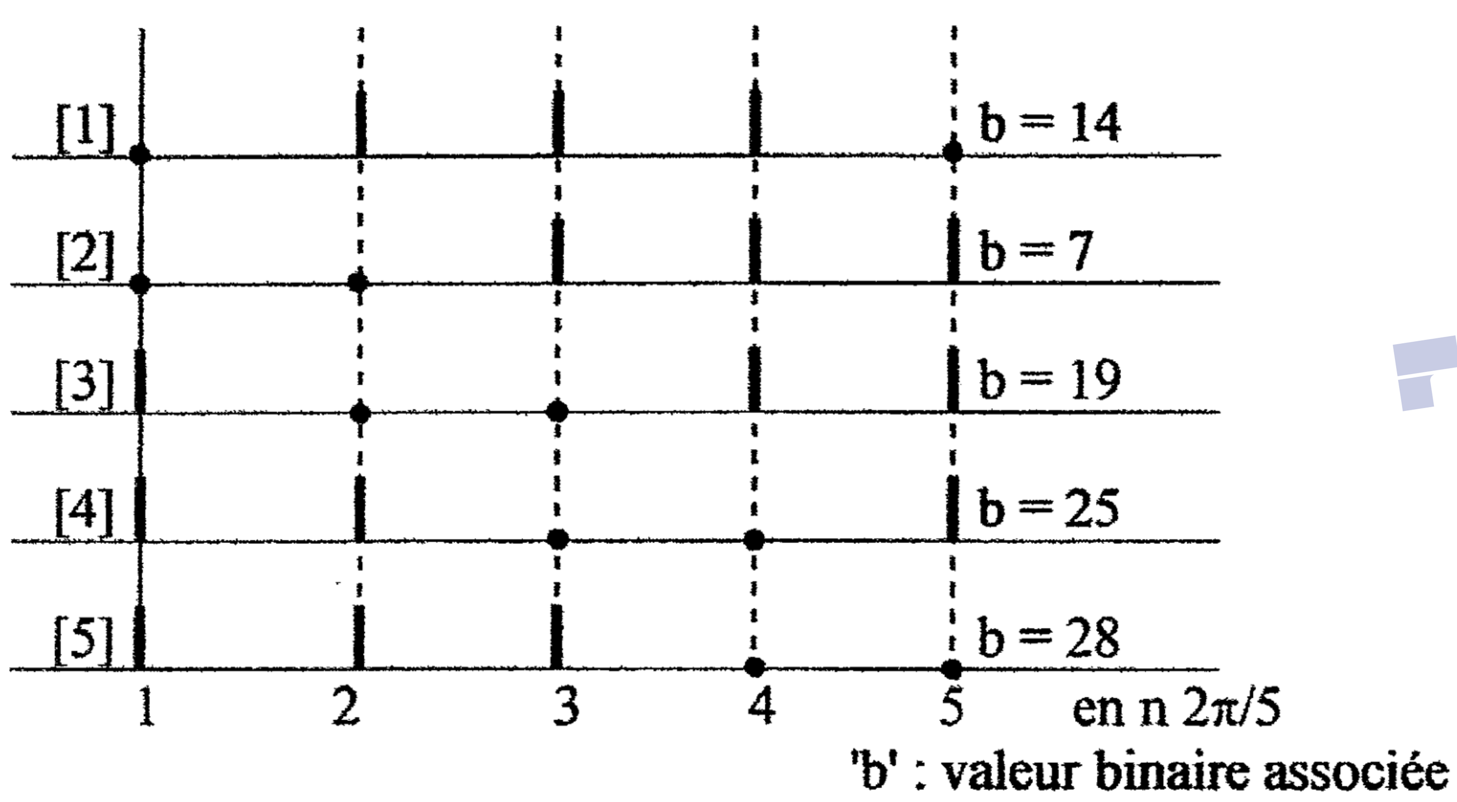


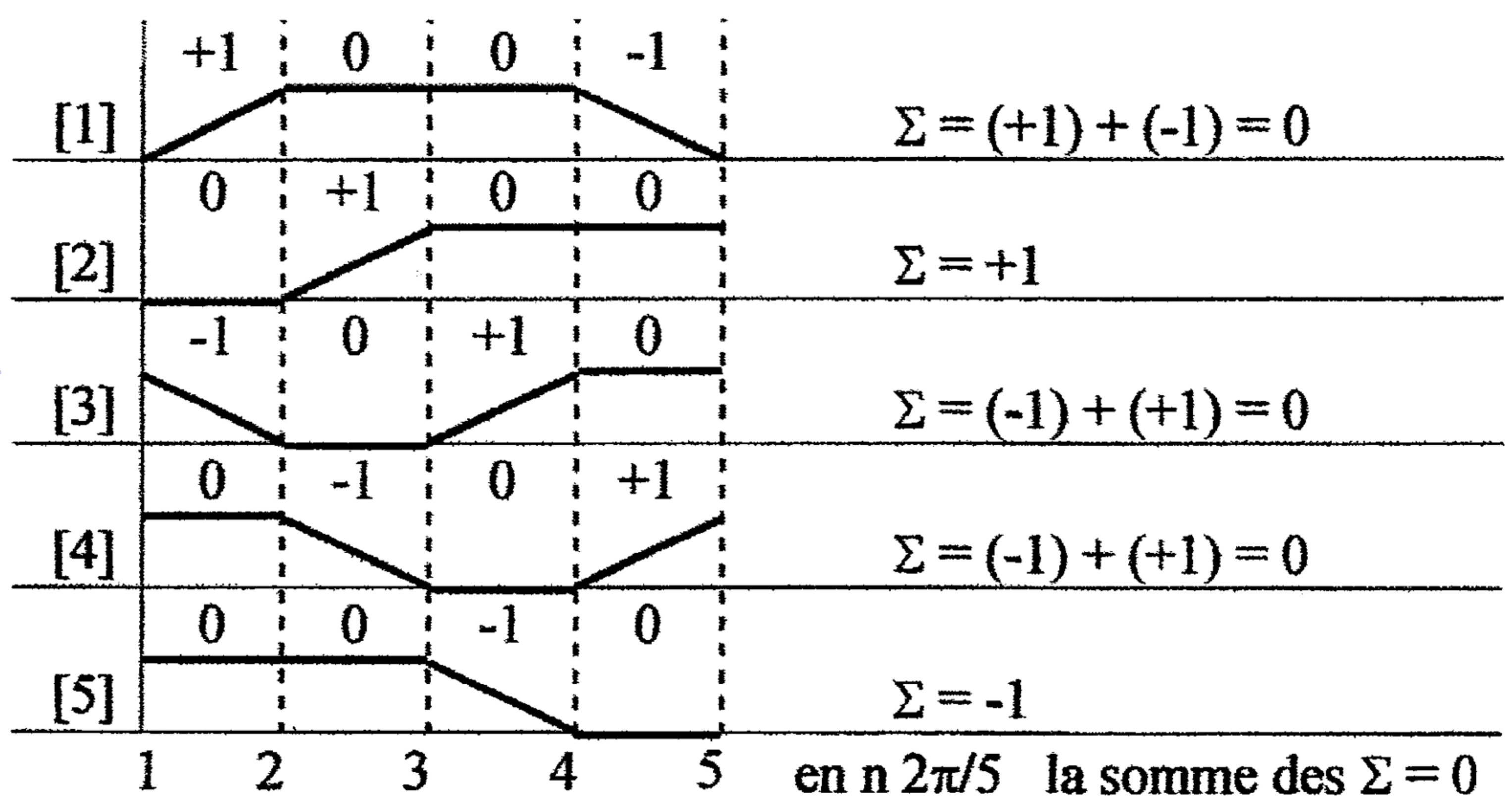
entre [ ] les éléments permutés. on remarque que chaque élément retrouve sa place initiale après trois permutations.[4]

- ***digression***

1 – mouvements des éléments

comme en théorie des systèmes microprogrammés, on peut établir le graphe d'évolution, ou timing-graph, de chaque élément ; les permutations étant discontinues, le timing offre une succession de points et de barres comme des *0* et des *1* en numération binaire. mais dans un système réel, le changement s'exprime par des montées et des descentes jamais instantanées qu'on image par des pentes logiques de valeurs *(+1)* et *(–1)* sur le graphe, *0* si rien ne change. nous aurons donc les deux représentations :





2 – mouvements des places

on peut aussi regarder du côté des places, les événements qui les concernent.

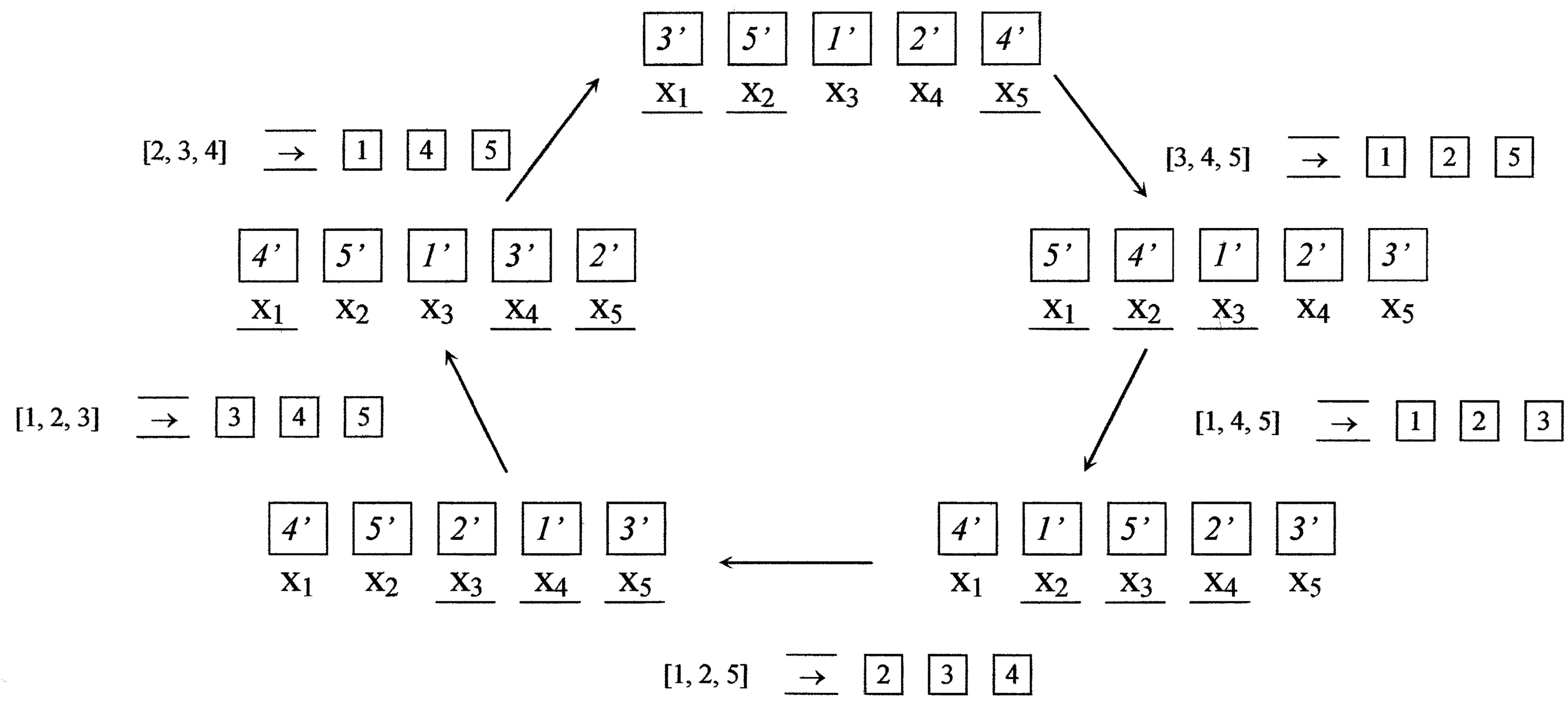
on choisit le mot de départ x_1 x_2 x_3 x_4 x_5. ici, la valeur des indices n'a rien à voir avec la valeur des éléments. il s'agit bien du numéro des places qui vont voir disparaître et apparaître les éléments de la permutation.

soit le mot de départ *3' 5' 1' 2' 4'* qu'en termes de places nous écrivons :

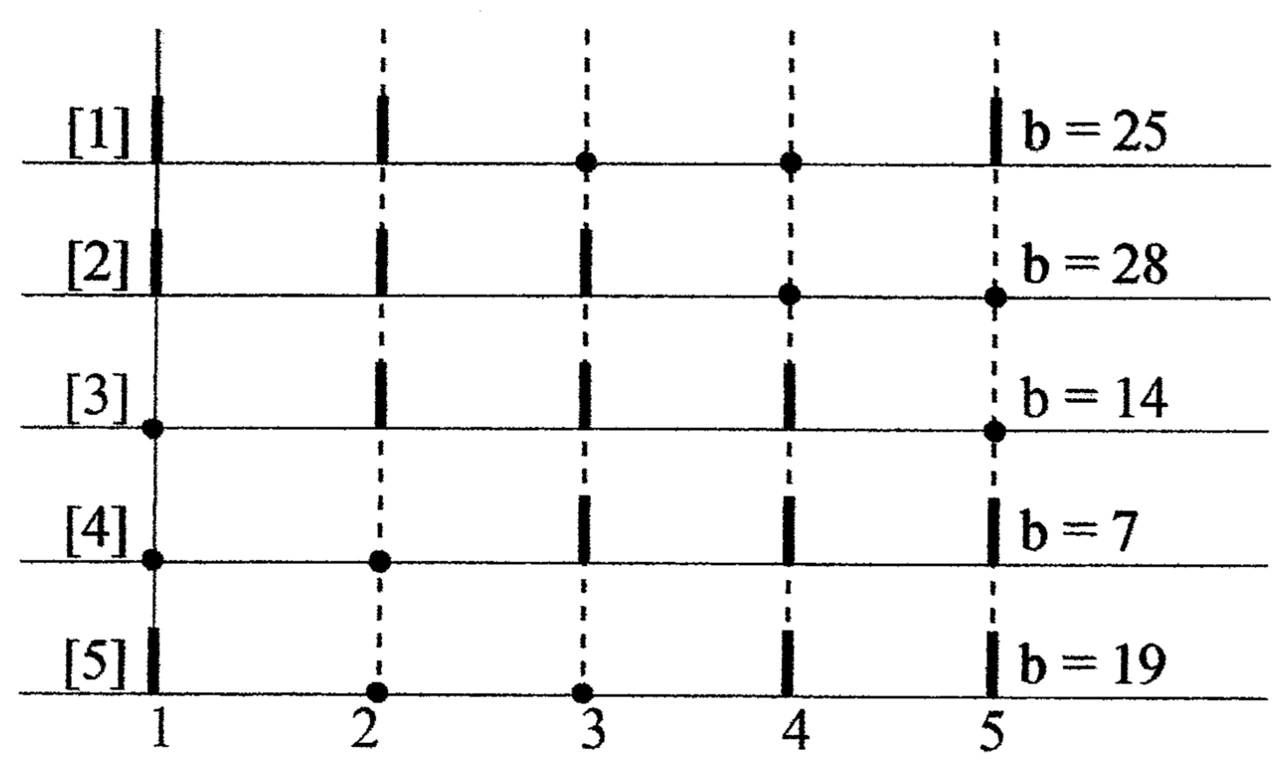
3'	*5'*	*1'*	*2'*	*4'*
x_1	x_2	x_3	x_4	x_5

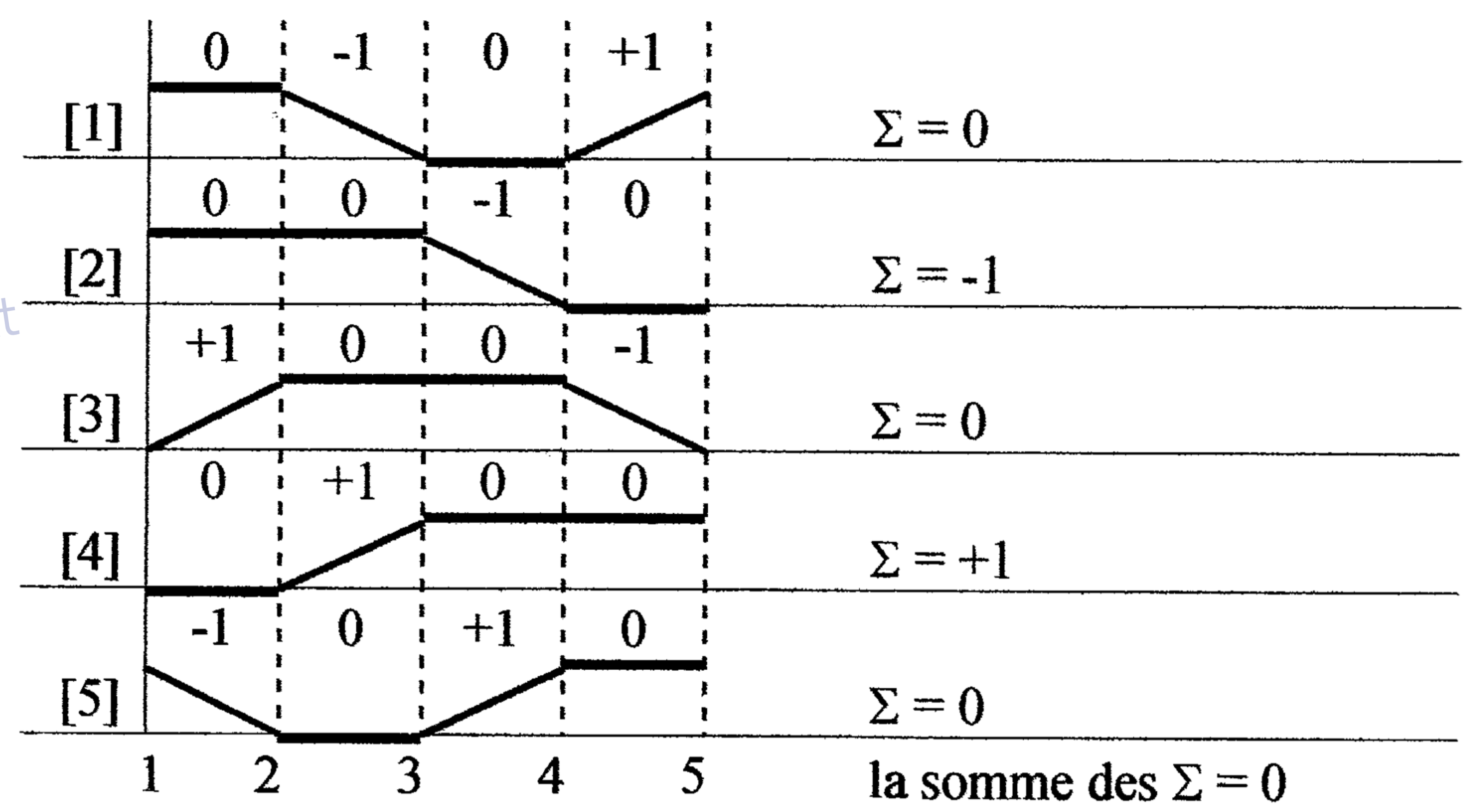
4 - il ne serait pas sans intérêt de comparer cette monstration des espaces E avec les *carrés magiques* — équivalents aux *ensembles différence parfaits* —, utilisés pour la construction de l'espace projectif $P^2(F_q)$, de penrose. cf. penrose **à la découverte des lois de l'univers**, odile jacob 2007, § 16.2 une géométrie finie ou infinie pour la physique, p349 et suite.

écrivons la succession des permutations en soulignant les places concernées.



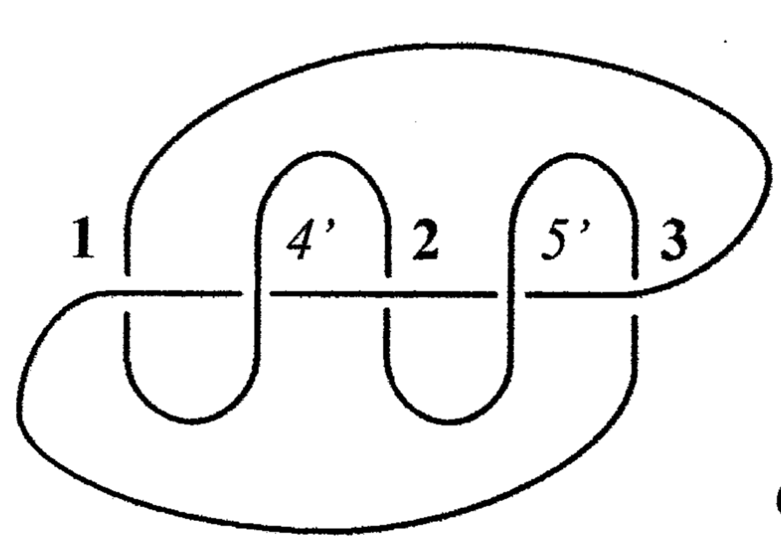
représentons le timing-graph des places en notant d'une barre un événement :

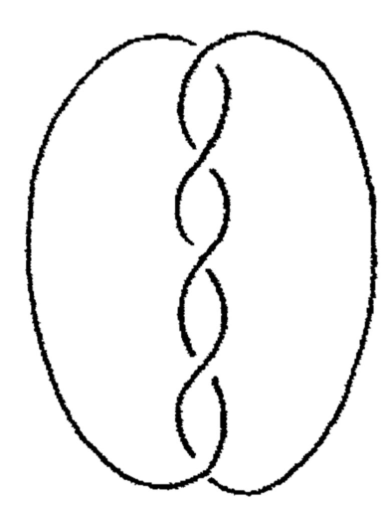




nous remarquons que les timing-graphs des places peuvent se coupler aux timing-graphs des éléments qui leurs ressemblent : ainsi $\boxed{1} \sim \boxed{4}$, $\boxed{2} \sim \boxed{5}$, $\boxed{3} \sim \boxed{1}$, $\boxed{4} \sim \boxed{2}$, et $\boxed{5} \sim \boxed{3}$.

formons le mot correspondant *4' 5' 1' 2' 3'*, soit le mot-nœud **1***4'***2***5'***3***1'***4***2'***5***3'* et dessinons le nœud correspondant :



c'est-à-dire le nœud-centre  .