Rapport de Projet

Sujet 2 :

Coloration pour les graphe de degré d avec d + 1 couleurs

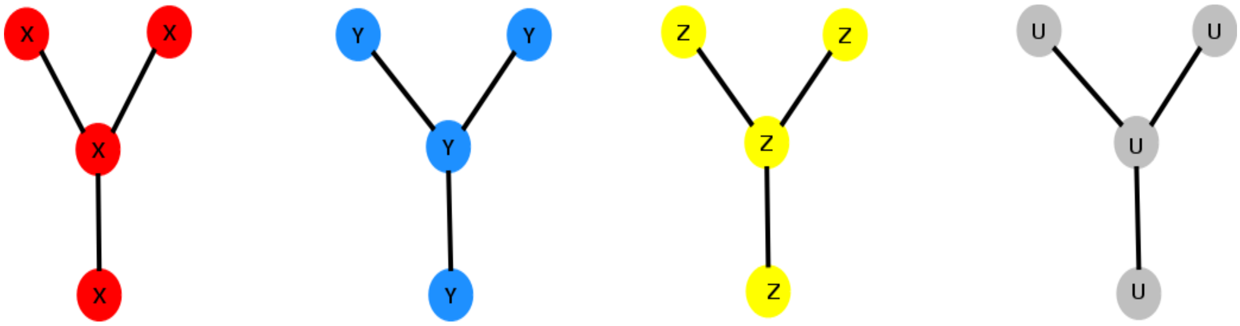
Master2 u-bordeaux1 S.D.R.P **Etudiants**: *Doan Chi-minh LouRuding*

Index

1. Un système de réécriture pour le cas d= 3

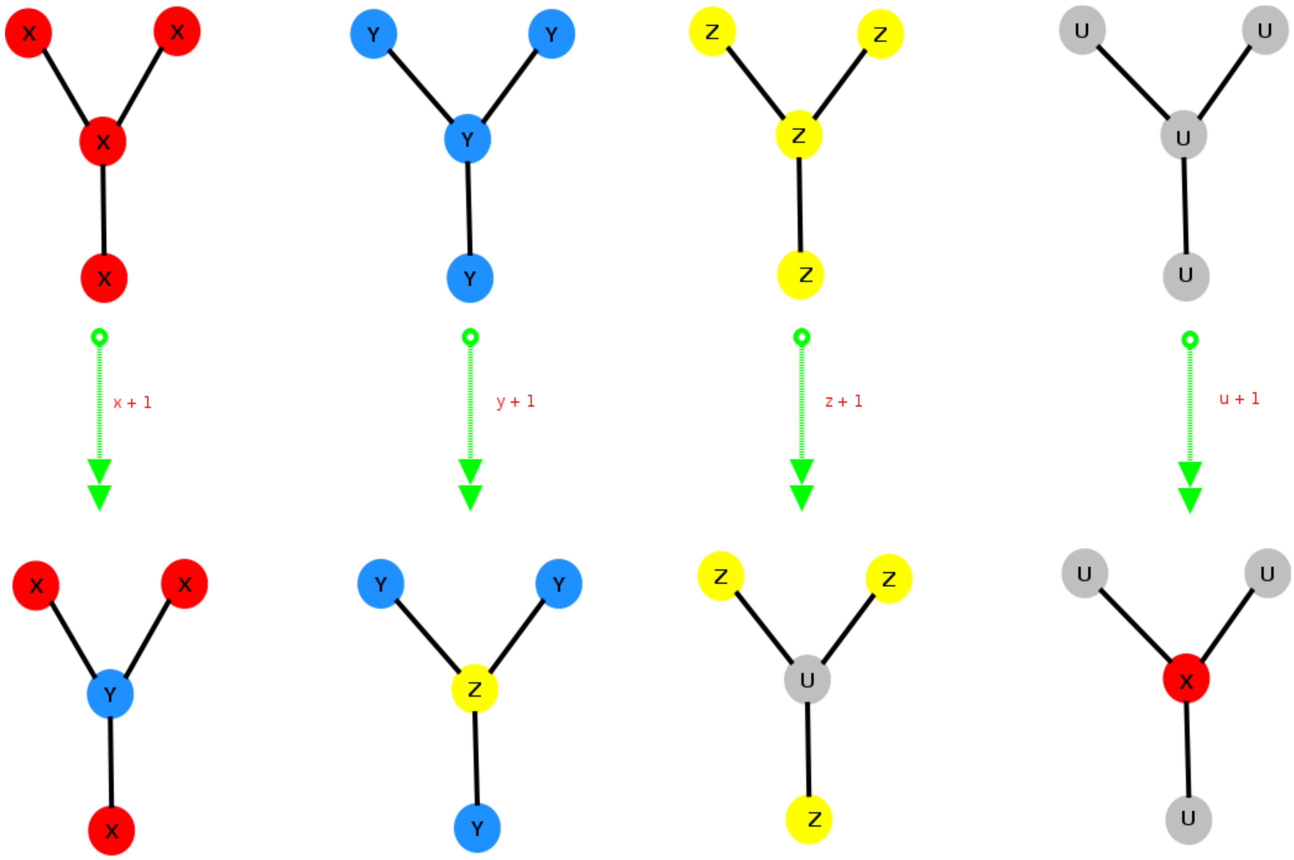
Pour ce cas-là on définit d + 1 couleurs pour la coloration, soit 4 couleurs. Je définis les 4 couleurs avec les lettres X , Y , Z, U. Au niveaux de algorithme, j'ai numéroté les quatre couleurs avec 0 , 1 , 2 , 3
pour X , Y , Z , U.

Le système de réécriture :



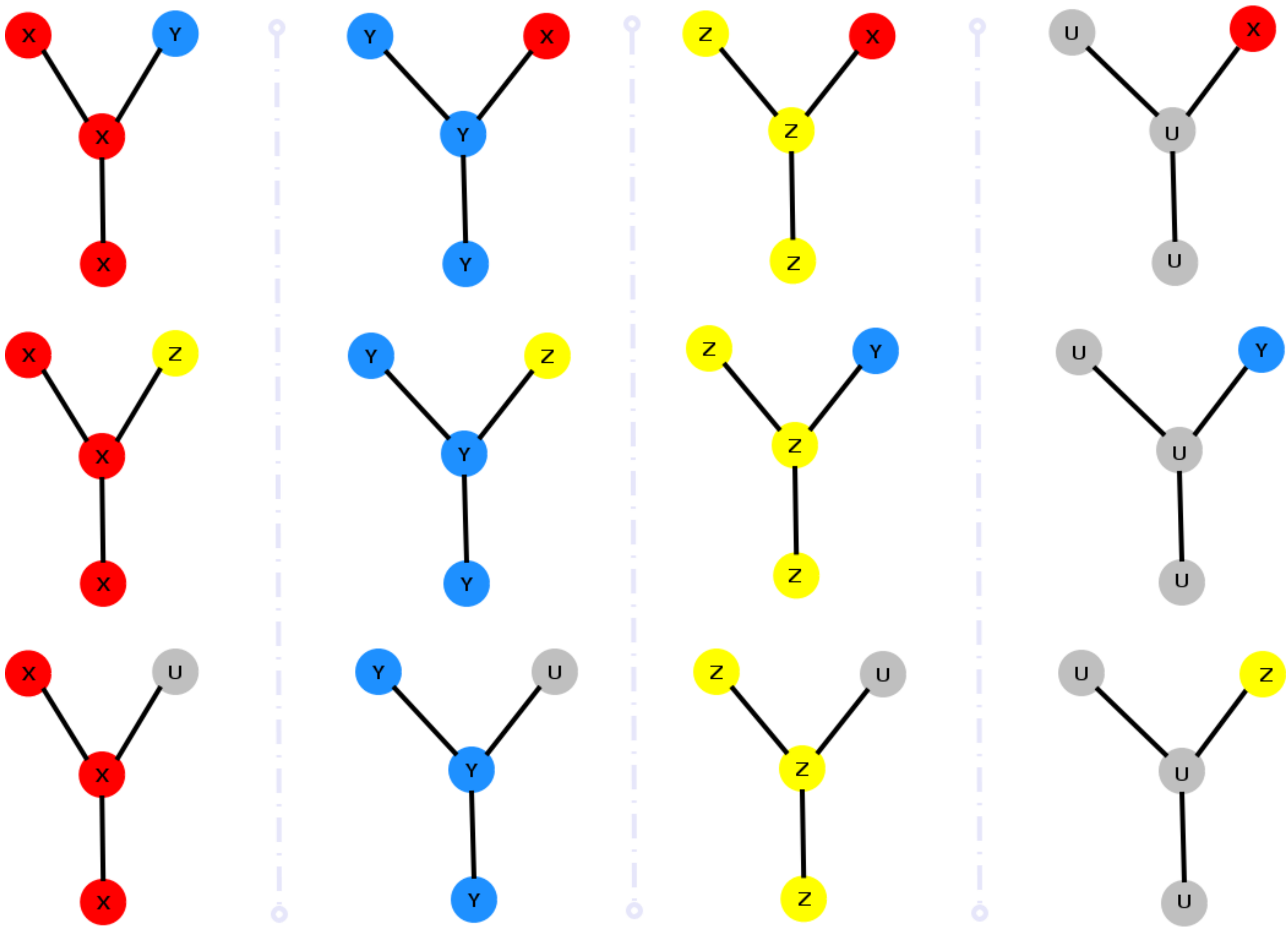
On en déduit que c'est la même possibilité , c'est-à-dire que tous les sommets d'une même boule ont la même couleur. Donc pour ce cas là il suffit de changer une fois la couleur du sommet au centre de la boule. Après cela, le sommet du centre de la boule peut avoir différente couleurs que celles de ses voisins.

Système de réécriture règle 1



Donc pour le premier cas, on peut juste changer la couleur du centre de chaque boule. Comme il n'y pas de priorité dans le changement de couleur, parce que si le sommet du centre change sa couleur, ça sera différente que ses voisins. Donc pour changer la couleur, au niveau du code, on ajoute 1 à l'ancienne couleur. Si on a X, on le changera en Y. Pour les autres on fait de même. Par contre quand on a la couleur U, on change en X. On aura alors **(couleur + 1) % 4** .

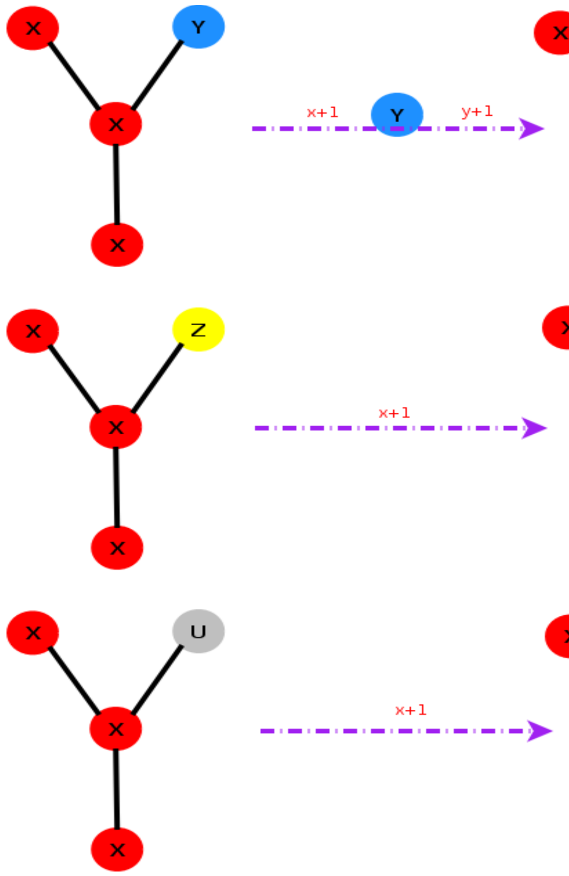
Dans l'autre cas, si il y a deux voisins qui ont la même couleur que le sommet du centre, on a le schéma suivant:

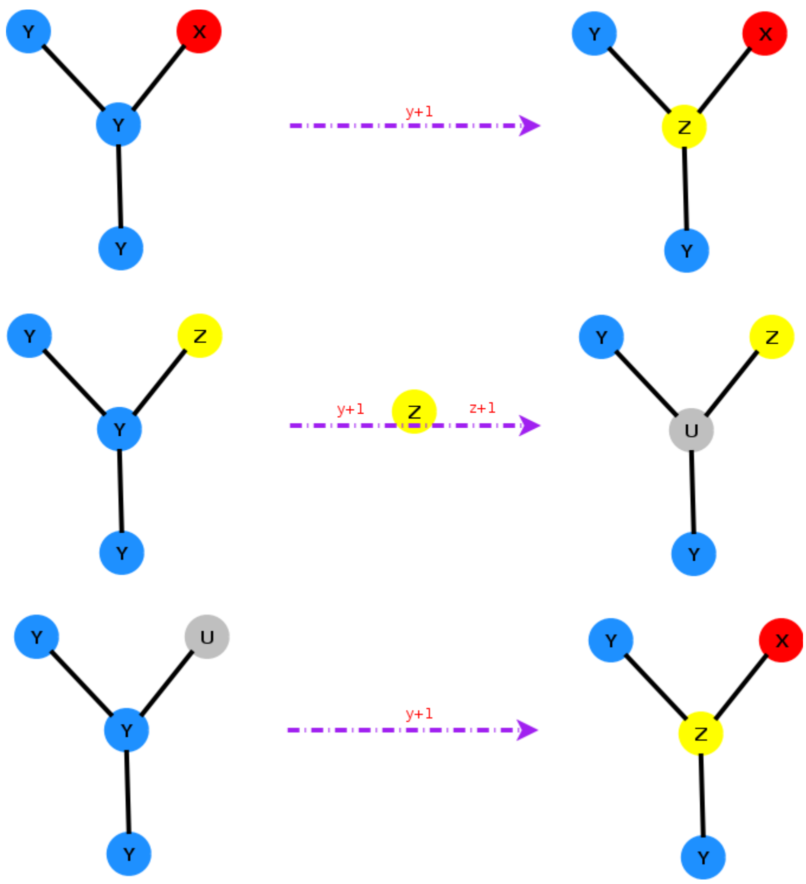


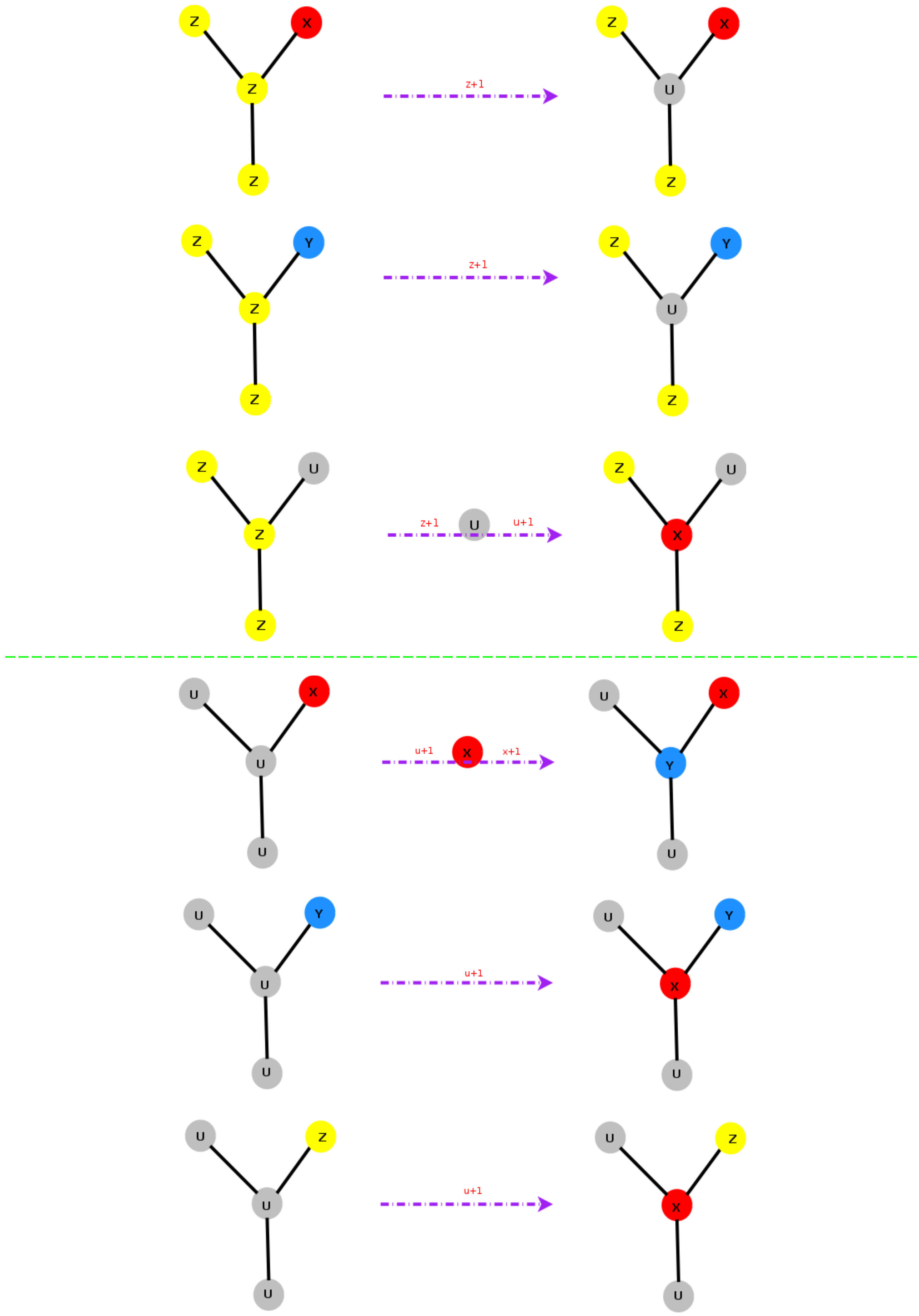
Ci-dessus, la figure représente le deuxième cas. Pour changer la couleur du sommet du centre d'une boule, il faut vérifier si après le changement de la couleur du sommet du centre, la couleur du sommet du centre est différente de celles des voisins. Donc on définit un système de réécriture pour ce cas:

Tournez la page pour avoir

Système de réécriture règle 2

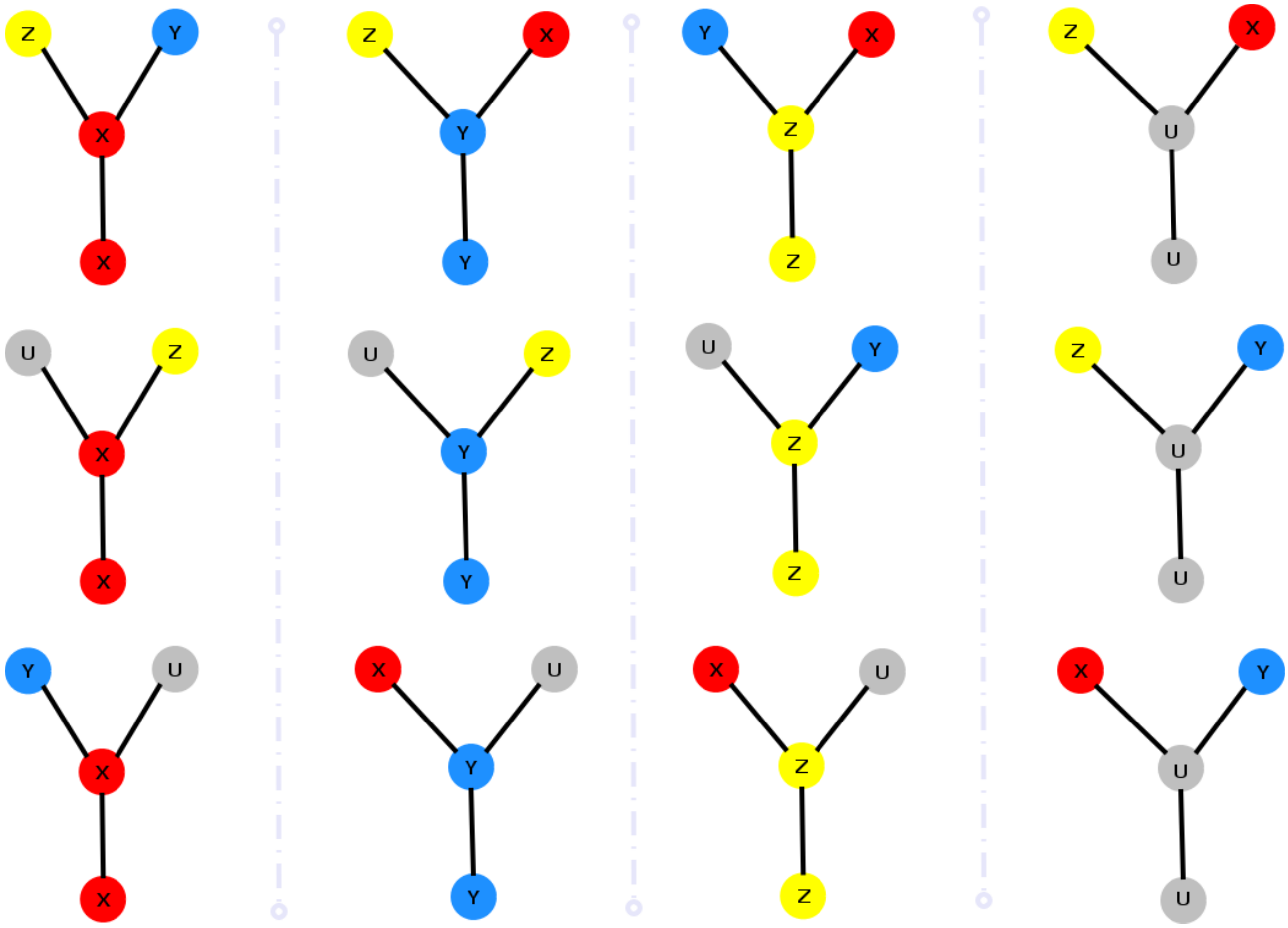

X
Y
X
X
x+1
Y
y+1
X
Y
Z
X
X
Z
X
X
x+1
X
Z
Y
X
X
U
X
X
x+1
X
U
Y
X


Y
X
Y
Y
y+1
Y
X
Z
Y
Y
Z
Y
Y
y+1
Z
z+1
Y
Z
U
Y
Y
U
Y
Y
y+1
Y
X
Z
Y


z+1
z+1
z+1 U u+1
u+1 X x+1
u+1
u+1

La figure ci-dessus représente le système de réécriture pour le deuxième cas. On s'aperçoit que si le sommet du centre change une fois sa couleur, sa nouvelle couleur n'est pas forcement différente de celles de ses voisins. En effet, elle peut être de la même la couleur que son troisième voisin dont l'ancienne couleur du sommet du centre était différente de la couleur. Donc il faut encore changer une fois la couleur, et si elle est différente de celles de tous les voisins, c'est correcte.

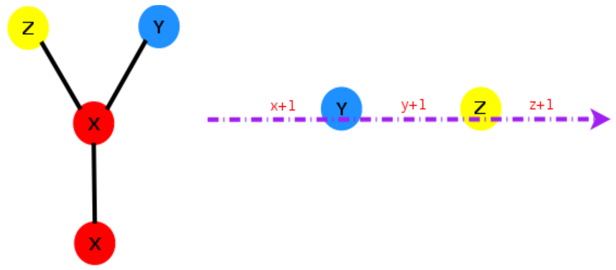
Dans le dernier cas, pour chaque boule il y a un seul sommet voisin qui a la même couleur que le sommet du centre :

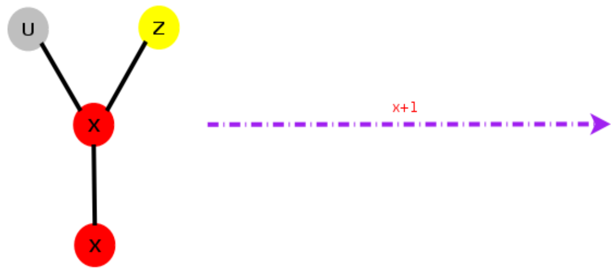


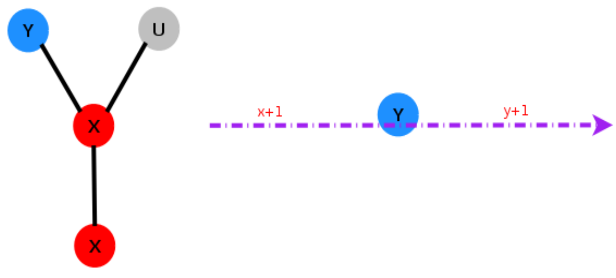
Dans ce cas, les trois voisins du sommet du centre de chaque boule ont des couleurs différentes, et celle du sommet du centre est la même que la couleur d'un de ses voisin. Ainsi quand le sommet du centre doit changer sa couleur plusieurs fois, si cela tombe sur sa couleur précédente qui est différente que celle d'un voisin, après le changement, sa nouvelle couleur est la même que celle l'autre voisin. Donc il faut changer sa couleur plusieurs fois jusqu'il a la une couleur dont aucun voisin n'a même.

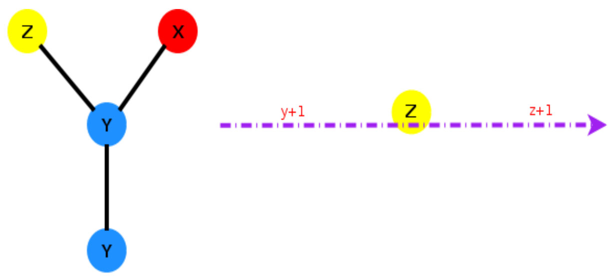
Voyons la page suivante qui présente le système de réécriture pour le cas ci-dessus.

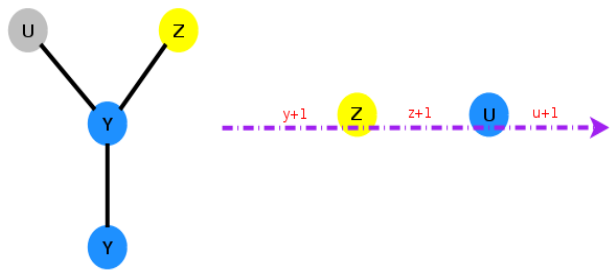
Système de réécriture règle 3

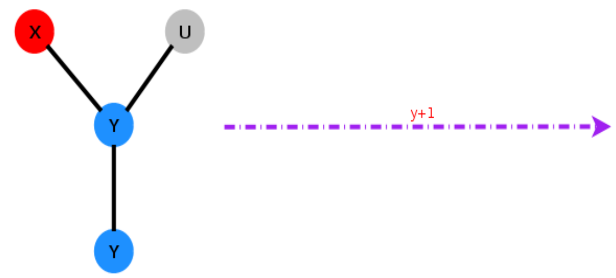

x+1
y+1
z+1

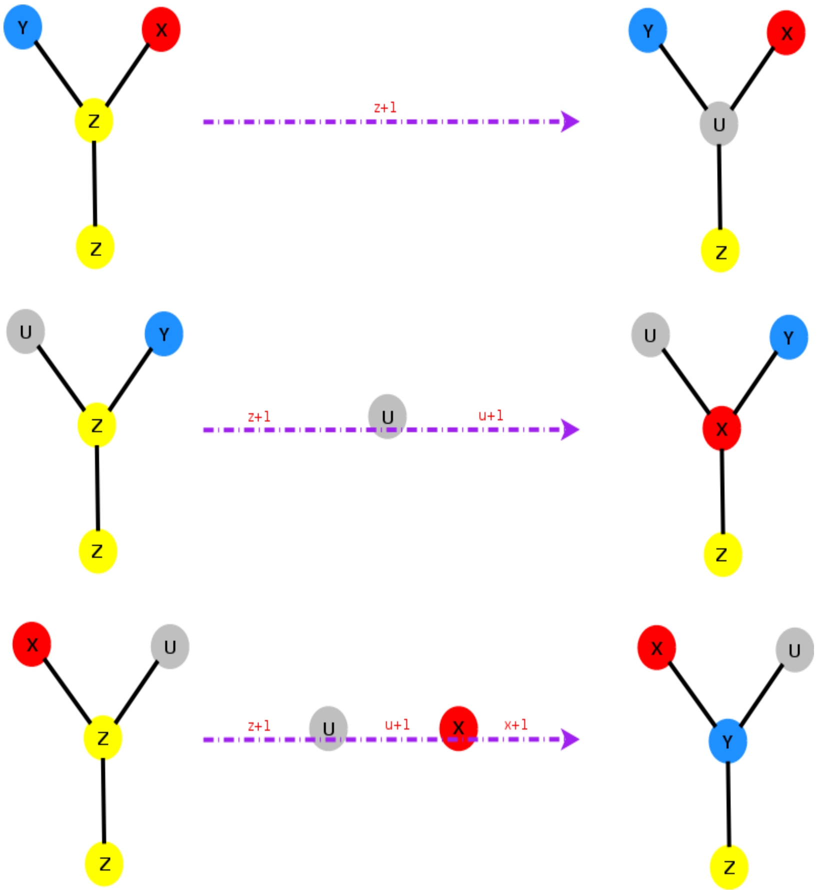

x+1

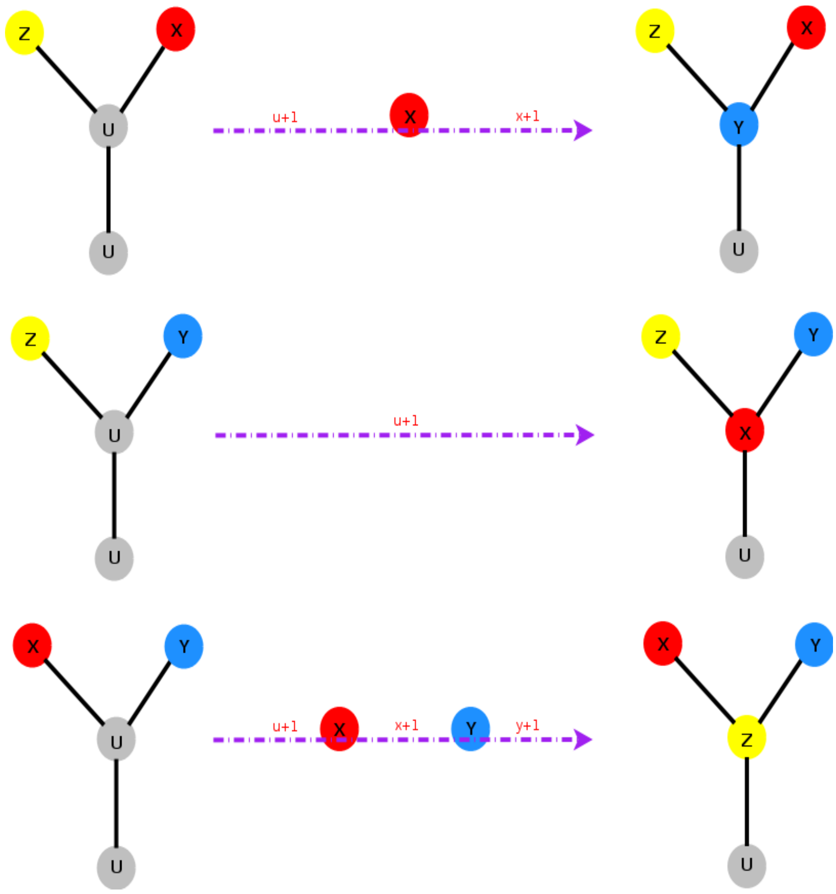

x+1
y+1


y+1
z+1


y+1
z+1
u+1


y+1


Y X Z Z
z+1
Y X U Z
U Y Z Z
z+1 U u+1
U Y X Z
X U Z Z
z+1 U u+1 X x+1
X U Y Z


Z X U U
u+1 X x+1
Z X Y U
Z Y U U
u+1
Z Y X U
X Y U U
u+1 X x+1 Y y+1
X Y Z U

La figure ci–dessus représente le troisième et dernier cas, parce que selon nos étapes, si il y avait encore des cas, cela signifierait que la couleur du sommet du centre d'une boule, serait différente de celle de ses voisins. Une fois qu'il commence comme ça , on peut dire que cette boule a finit sa coloration local.

2. Commentaire sur le système de réécriture pour le cas d= 3

Ce système colorie tous les sommets des graphe de degré 3 . Au niveau du code, on laisse les sommets envoyer des messages à leurs voisins, et puis ces sommets reçoivent alors les couleurs de leurs voisins. Ainsi, on essaie de trouver une couleur différente à lui affecter. Alors le sommet est correcte.

Ensuite, ce système de coloration est bien convergent. D'après le système de réécriture que l'on a présenté dans le premier chapitre, on voit bien que les trois règles convergent dans l'ordre. Chaque règle considère un cas où il y a moins de sommets voisins qui ont la même couleurs que celle du sommet du centre d'une boule, en opposition la règle précédente.

La règle 1 traite le cas où trois voisins de chaque sommet de centre ont la même couleurs que celui de centre.

Et ***la règle 2*** traite le cas où il n'y a que deux voisins qui ont la même couleur.

En fin ***la règle 3*** traite le cas où il ne reste qu'un voisin qui a la même couleur que celle du sommet de centre.

Si il y a encore un autre cas qui veut utiliser le système et il ne trouve aucune règle qui lui convient, je peut dire que ce cas est le cas qu'on veut. C'est-à-dire que aucun voisin n'a la même couleur que celle du sommet de centre de boule.

On peut bien croire que ce système converge .

Aussi on peut faire une formule qu'on a vu en cours. On a un ensemble de (S , >), et f une d'une fonction pour colorier le graphe G, et donc chaque sommets sont colorié avec un ensemble de label λ. , donc on peut avoir

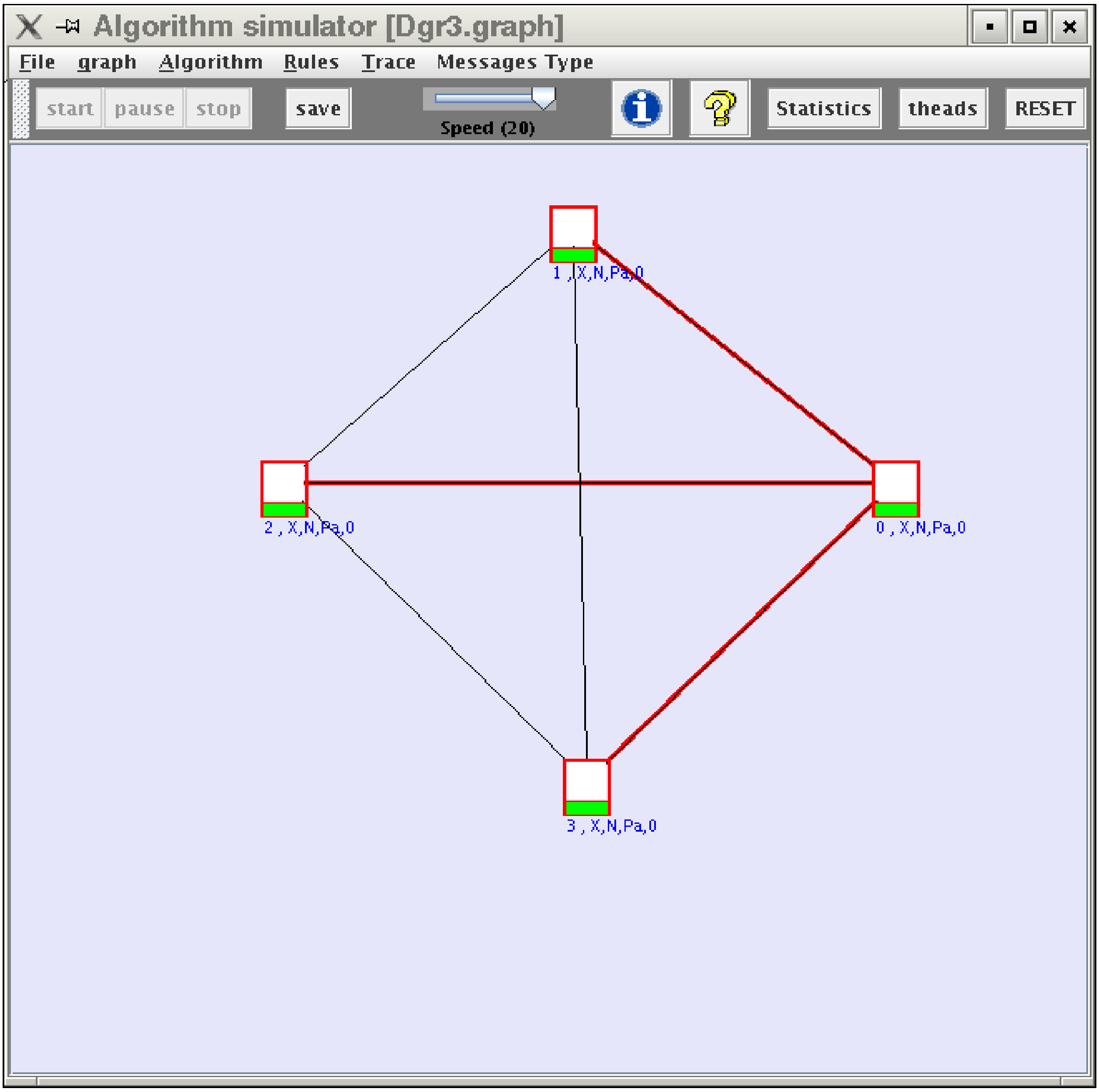
$(G, \lambda) \to (G, \lambda')$ alors $f(G, \lambda) \to f(G, \lambda')$

C'est **noethérien** .

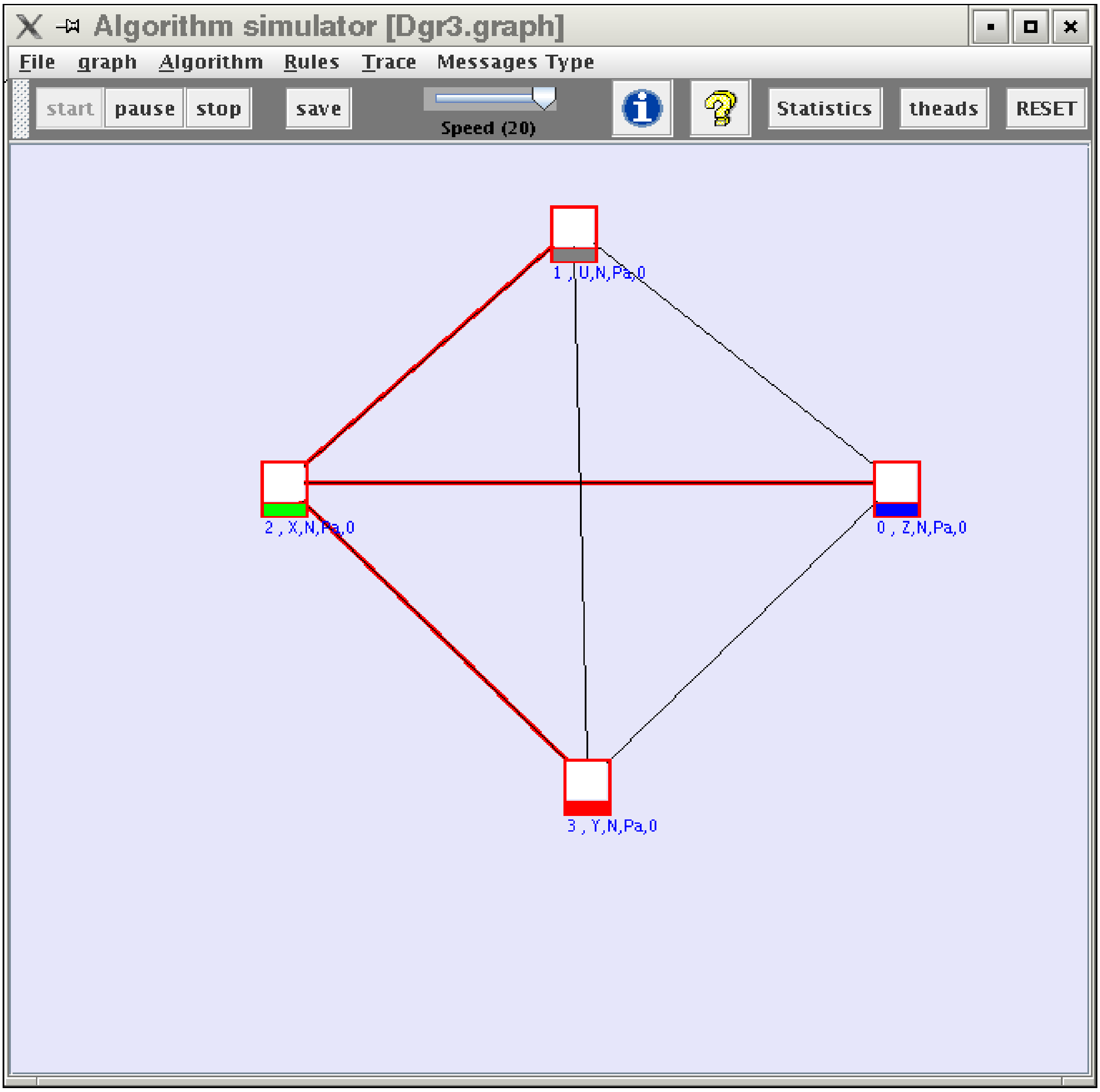
3 . Implémentation sur logiciel VISIDIA le système de coloration du graphe de degré 3

Dans ce chapitre je vais faire quelques présentation de l'implémentation pour ce système grâce à logiciel *VISIDIA.*

D'abord je teste un graphe simple ayant 4 sommets de degre 3. Au début de lancement de l'algorithme :

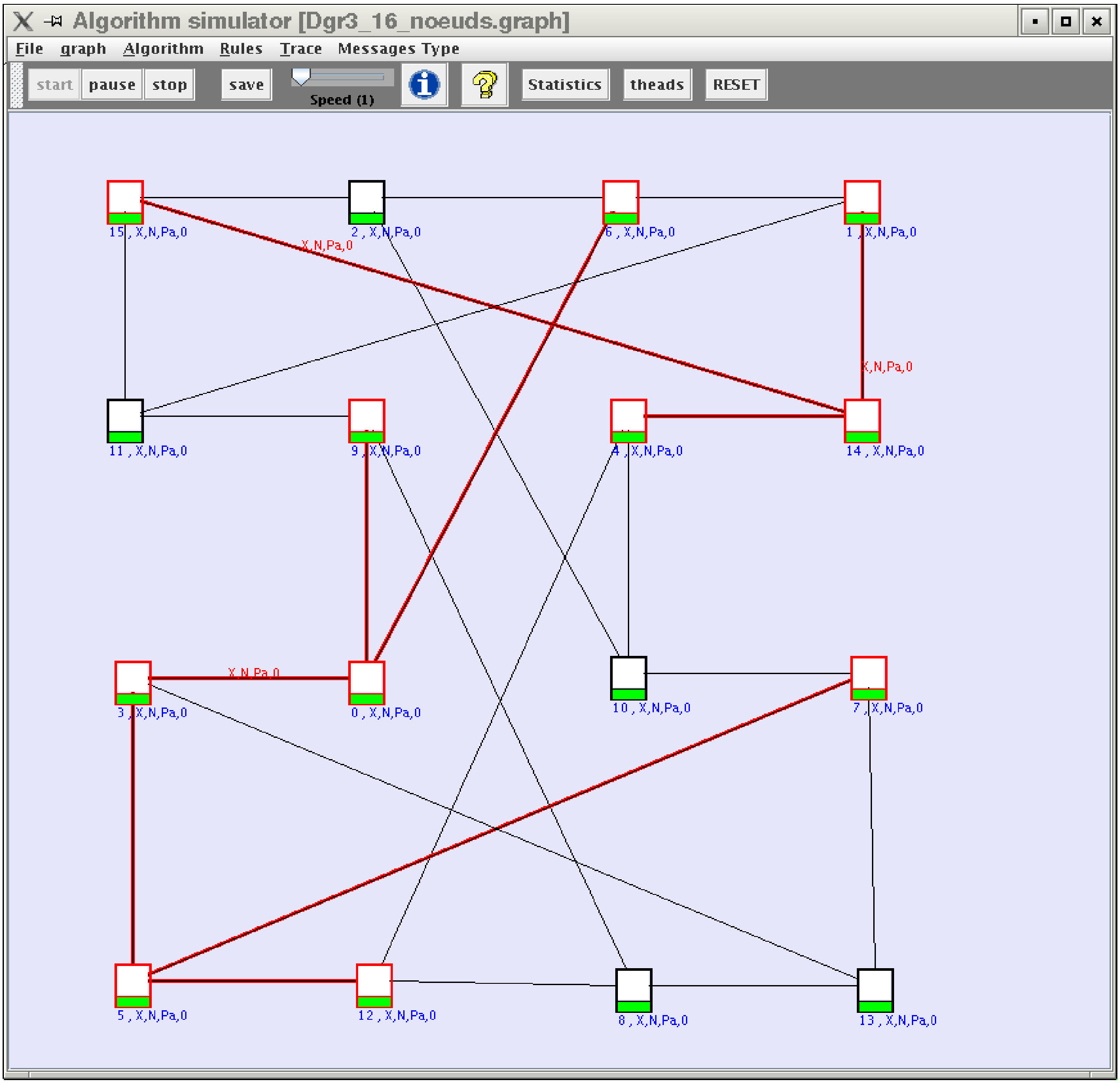


À la initiation on voit bien que les sommets sont initiés avec des couleurs identique. Et après que l'algorithme tourne, et tous les sommets cherchent les couleurs de leurs voisin et s'affecte la différente couleurs, j'obtiens un résultat correcte . Tournez la page pour voir ce que ça donne. S.V.P.



Je vois bien que les sommet ont leurs couleurs différentes que celles de leurs voisins.

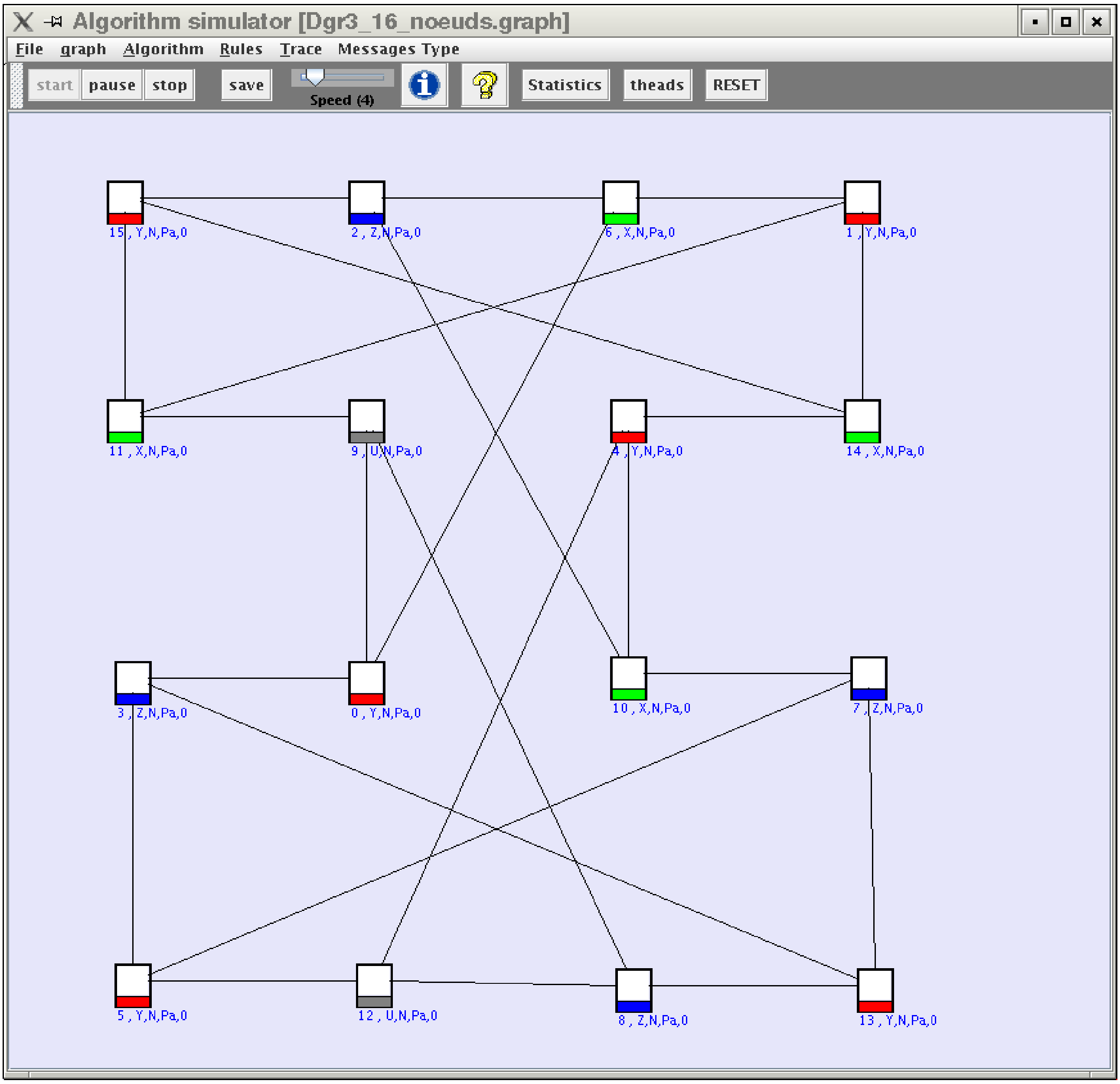
Ensuite je tester ce système avec un graphe de 16 sommets,Voyons le graphe en initiation.



Je vois bien il y a trois boules qui a commencé tourner les règles.

Et après tourner l'algorithme ça donne le bon résultat aussi .

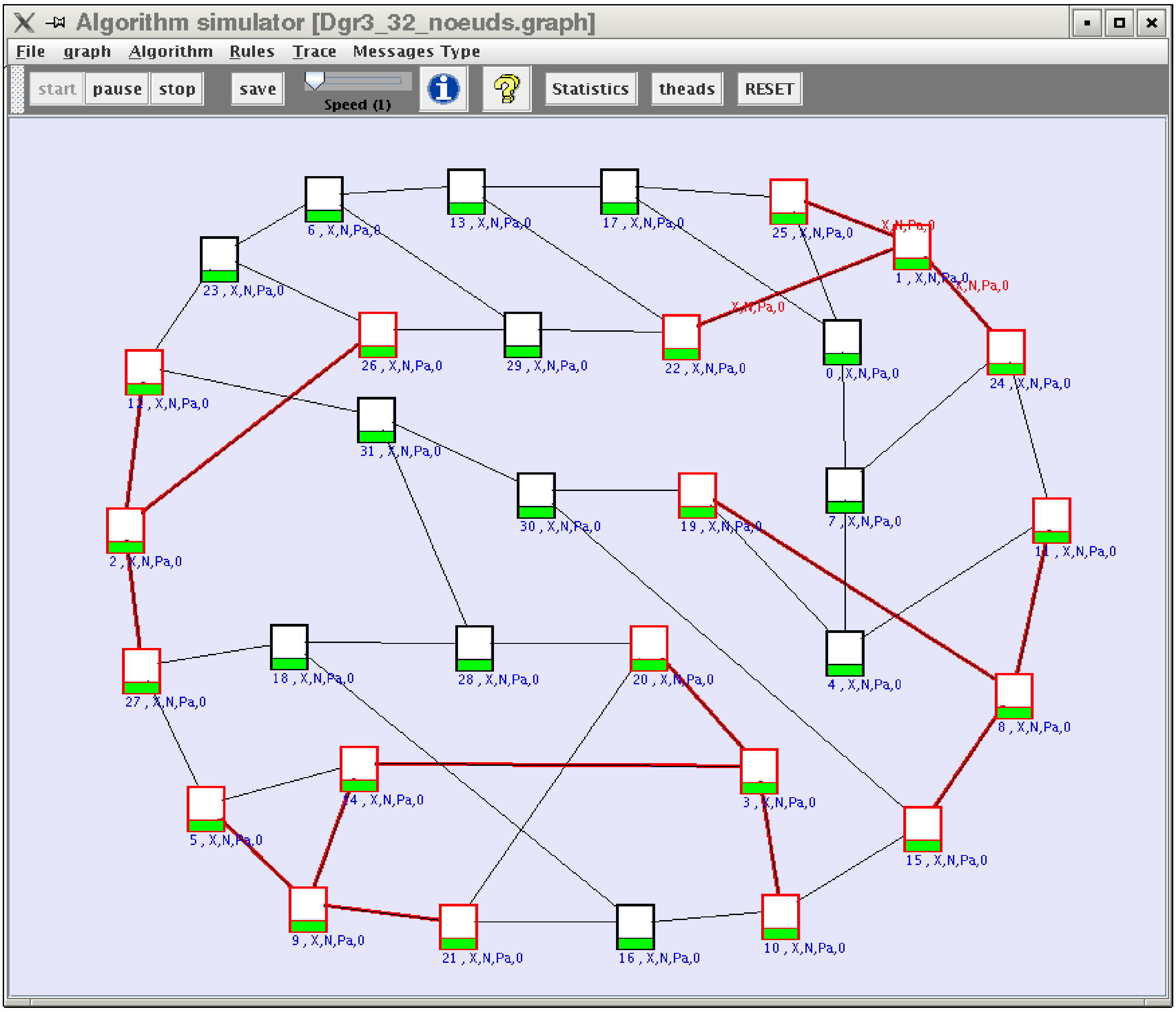
Voyons, tournez la page S.V.P



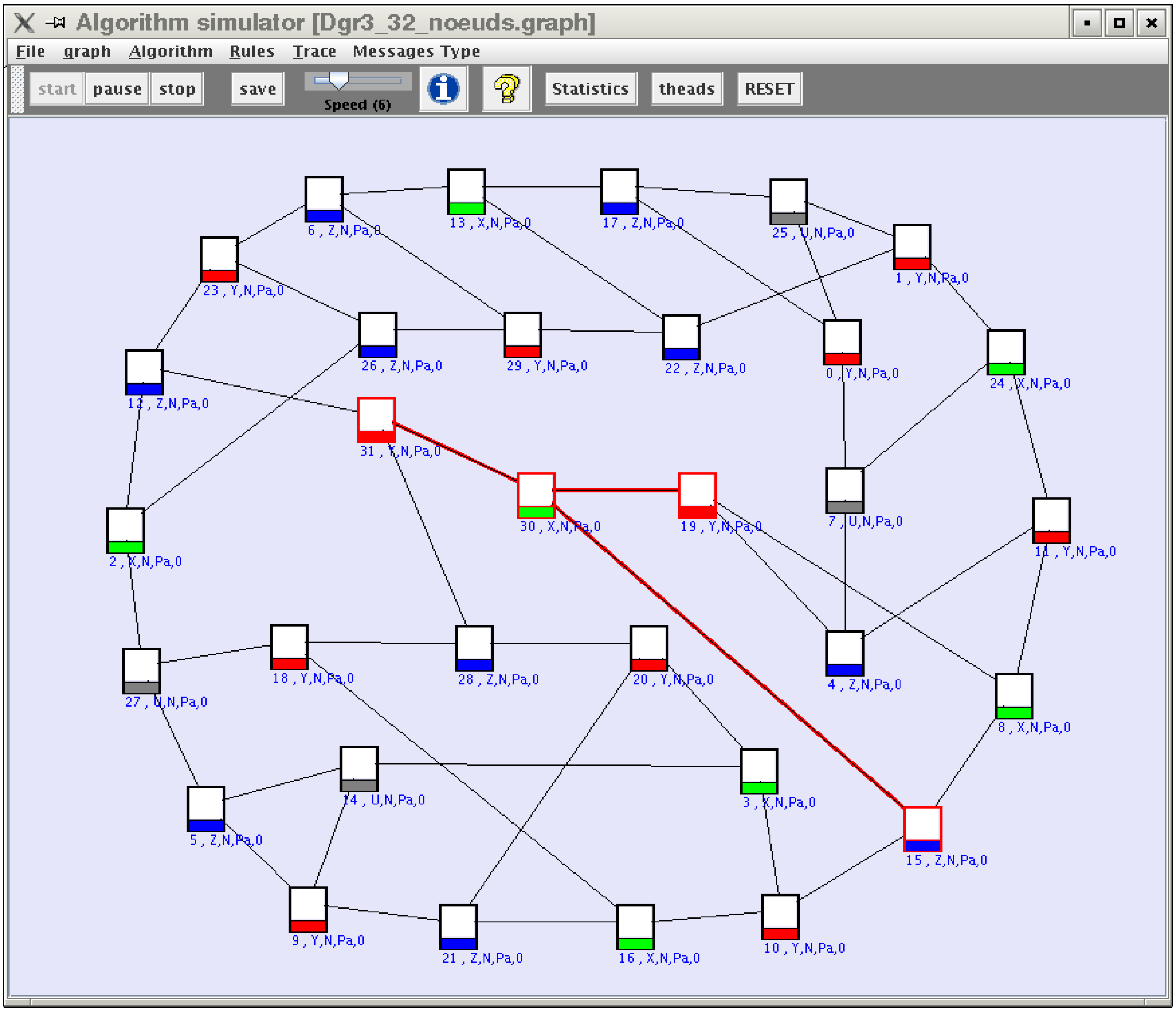
Avec ce système de coloration , tous les sommets ont la différente couleur que les sommet du tour de boule.

Et en fin j'ai fait un teste avec un graphe de 32 sommets , donc dedans il peut y avoir beaucoup plus de boules .

Tournez la page pour voir l'initialisation de implémentation avec VISIDIA.

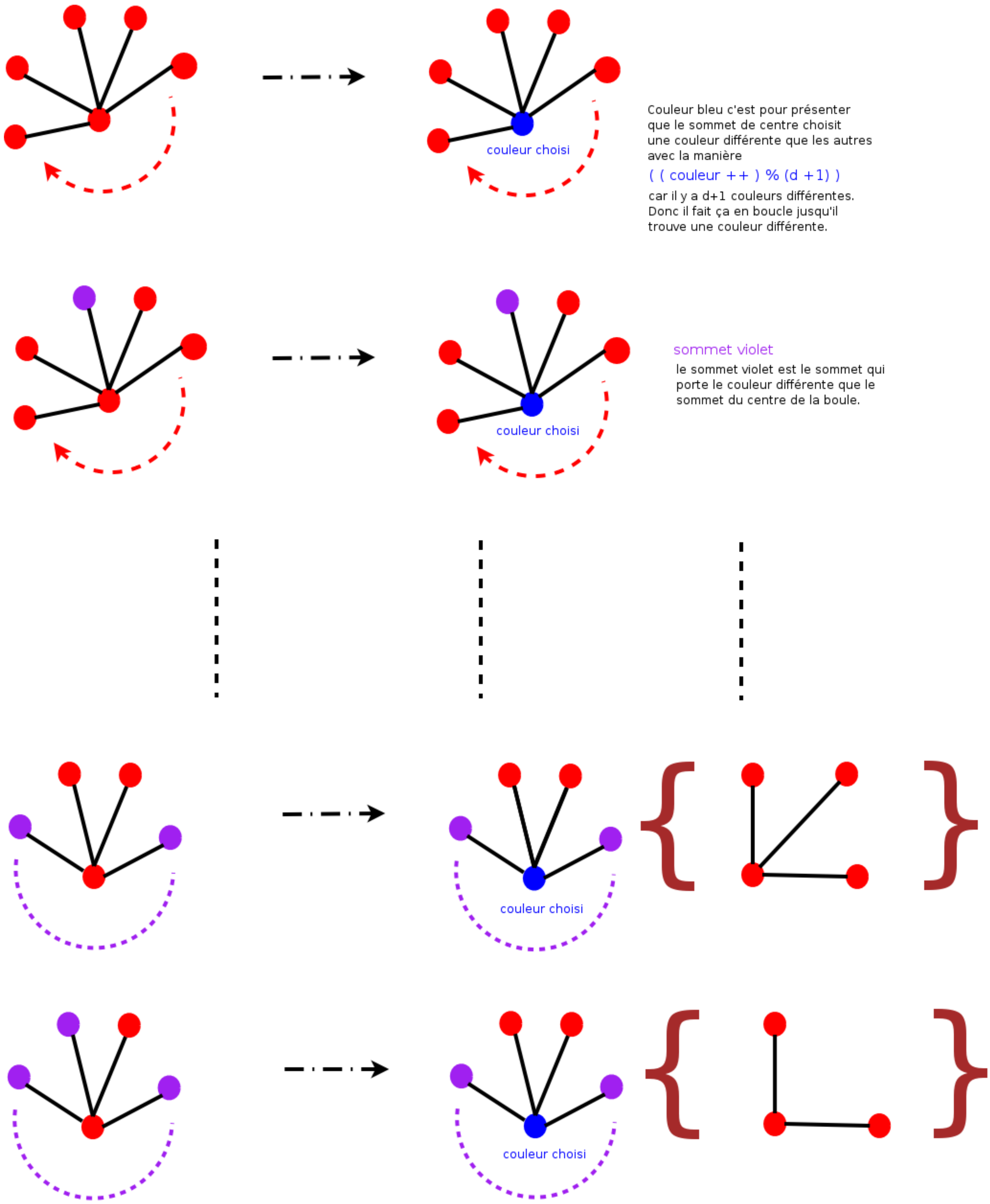

Algorithm simulator [Dgr3_32_noeuds.graph]
File graph Algorithm Rules Trace Messages Type
start pause stop save Speed (1) Statistics theads RESET

et après l'implémentation de coloration.



Et cette fois-ci ça donne bon résultat aussi . Tous les boules sont bien coloriées.

4. Proposition un système de réécriture pour les de degré quelconque



La figure ci–dessus représente le système de réécriture pour le cas d quelconque.. Dans le schéma, le sommet du centre change sa couleur à bleu, ici bleu est une couleur pour généraliser la couleur que le sommet du centre trouve à la fin, elle différente que celles des voisins.

Et autour les voisins ont la couleur violet, la couleur violet représente les couleurs qui sont différente que celles du sommet du centre.

Pour que le sommet du centre trouve la couleur différente, je pense que l'algorithme est analogue (semblable) que l'algorithme de système pour les graphe de degré 3.

En initiation, on définit chaque sommet de centre d'un boule v, tous les voisins d'une boule w .

```
v envoyer message aux tous les  w
v reçoit les couleurs des w
while ( il existe un w qui a la même couleur que v)

        v change un coup sa couleur( couleur ++)

End while
```