DSRT0703 - Proposition de correction

Questions de cours

1. Anneau

Graphe non orienté, connexe et régulier de degré 2.

2. Problème d’exclusion mutuelle

Utilisée quand un ensemble de sites a besoin d’accéder à une ressource. Durant l’exclusion mutuelle, un seul site au plus peut être en section critique à un instant donné. S’il essaye de rentrer en section critique il finira par y arriver. L’algorithme doit être capable de faire passer un site de l’état « entré » à l’état « section critique » en vérifiant les propriétés de sûreté et de vivacité.

3. Algorithme de parcours

Pas encore fait

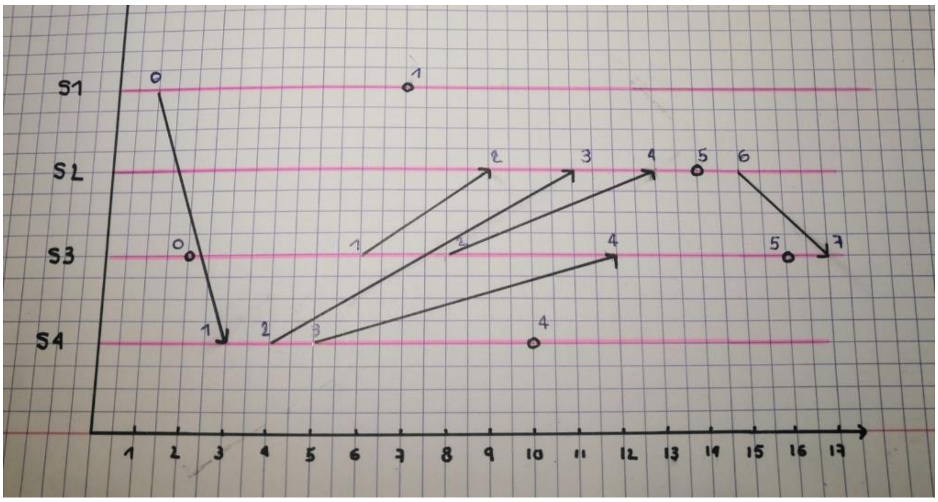
4. Algorithme d’exclusion mutuelle de Lelann (jeton circulant sur un anneau)

```
reception_jeton()
	si etat == E alors
		etat = SC
	sinon
		envoyer_jeton(site_suivant)
	fin si
	etat = s
	envoyer_jeton(site_suivant)
fin reception_jeton
```

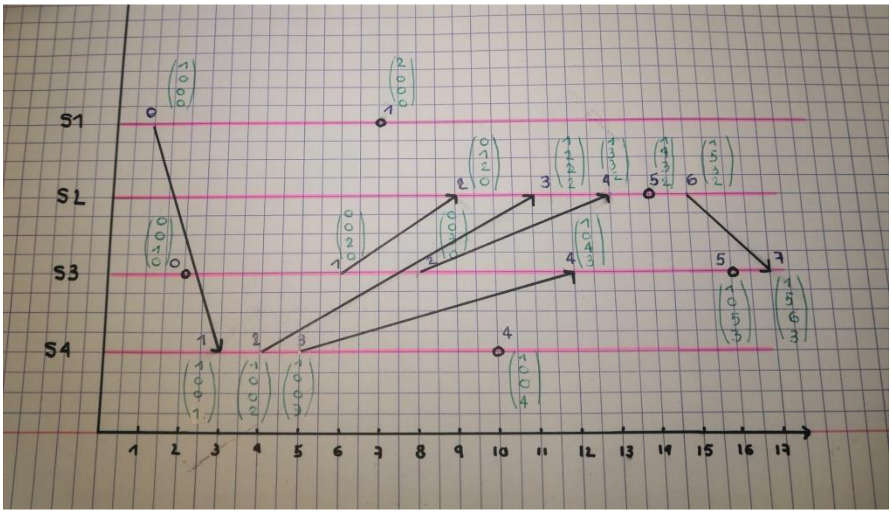
Causalité

Soit les évènements constituant l’execution (S) d’un système distribué :
E_1 : 1 envoie à 4, E_2 : interne 3, E_3 : 4 reçoit de 1, E_4 : 4 envoie à 2, E_5 : 4 envoie à 3,
E_6 : 3 envoie à 2, E_7 : interne 1, E_8 : 3 envoie à 2, E_9 : 2 reçoit de 3, E_{10} : interne 4,
E_{11} : 2 reçoit de 4, E_{12} : 3 reçoit de 4, E_{13} : 2 reçoit de 3, E_{14} : interne 2, E_{15} : 2 envoie à 3,
E_{16} : interne 3, E_{17} : 3 reçoit de 2.

1. Pour chaque événement, donner les horloges de Lamport



2. Pour chaque événement, donner les horloges vectorielles



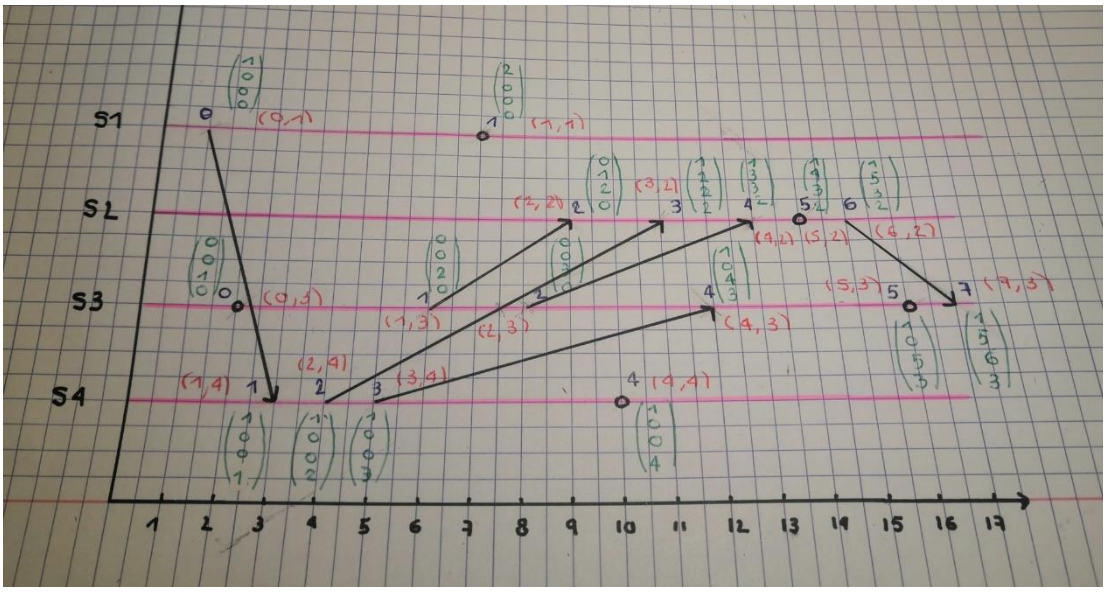
3. Donner une exécution équivalente en **maximisant** le nombre de messages en cours d'acheminement

Il faut se rappeler de toujours garder l'ordre causal : sur un même site l'ordre est strictement ordonné et l'événement d'envoi d'un message précède toujours la réception de ce même message. Le but ici est d'avoir **le plus** de messages en transition possible (d'avoir des vecteurs plus longs) Nous pourrions donc avoir :

E1 – E7 – E2 – E6 – E7 – E8 – E3 – E4 -E5 – E10 - E9 - E11 - E13 – E14 – E15 - E12 – E16 – E17

4. Ordonner totalement les événements

Il faut estampiller chaque événement (en rouge) ce qui donne :



Ainsi il est plus facile de les trier :

E1 – E2 – E7 – E6 – E3 – E9 – E8 – E4 – E11 – E5 – E13 – E12 – E10 – E14 – E16 – E15 – E17

5. Les événements E_1 et E_{17} sont-ils liés causalement ? Justifier

Vérifions si un des 3 points de la définition est vérifié :

- E1 et E17 sont 2 événements sur des sites différents, donc non.
- E1 n'est pas l'émission de E17, donc non.
- Vérifions enfin la transitivité : E1 et E3 sont liés causalement par le .2 de la définition du cours. E3 et E4 sont liés causalement par le .1 du cours. E4 et E11 sont liés causalement par le .2 du cours etc. On en conclut donc que par transitivité les événements E1 et E17 sont liés causalement.

6. Montrer que …

Un évènement $e = (c, m, d)$ est dit applicable sur $\gamma = (e_1, e_2, \dots, e_p, \dots, e_n, M)$ si $\exists p | e_p = c$ et $e(\gamma) = (e_1, e_2, \dots, d, \dots, e_n, M')$ avec $M' = M$ si e est un évènement interne, $M' = M \cup \{m\}$ si e est un évènement d'envoi et $M' = M \setminus \{m\}$ si e est un évènement de reception. Soit α une configuration et e_p et e_q 2 évènements sur 2 sites distincts. Ces évènements sont tous les deux applicables sur α. Montrer que $e_p(e_q(\alpha)) = e_q(e_p(\alpha))$ (2pts).

Juste une idée, mais pas aboutie je pense…

Ce sont 2 événements sur 2 sites distincts donc il n'y a pas de relation de causalité entre les deux. Nous pouvons donc réordonner les événements puisque l'ordre causal n'est pas changé.

Déroulement d'un algorithme

Dans l'algorithme 1, on considère un jeton qui circule et qui contient une table de n entiers (jeton.table) et un nombre de saut (jeton.hops).

Algorithme 1

Variables :
$etat = \{demande, SC, sortie\}$
Instructions
Reception jeton de j
$jeton.hops \longleftarrow jeton.hops + 1$
if $etat = demande$ **then**
  $etat \longleftarrow SC$
  Section Critique
  $etat \longleftarrow sortie$
end if
Choix $k \in Vois | jeton.table[k] = \min_{l \in Vois}\{jeton.table[l]\}$
$jeton.table[k] \longleftarrow jeton.hops$
Envoi jeton à k

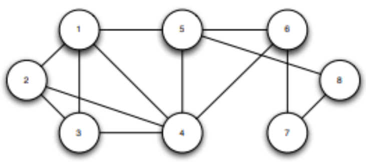


FIGURE 1 – Topologie du système distribué

1. Que fait cet algorithme ? A quelle classe d'algorithmes appartient-il ?

Cet algorithme utilise une stratégie de déplacement du moins récemment visité. Il est fondé sur l'unicité d'un jeton.

2. Déroulez cet algorithme en considérant que le jeton est détenu par le site 1, que la table du jeton soit initialisé à des valeurs nulles et que jeton.hops = 0

Jeton.tab :

Site 1	*Site 2*	*Site 3*	*Site 4*	*Site 5*	*Site 6*	*Site 7*	*Site 8*	*Jeton.hops*
0	0	0	0	0	0	0	0	0
0	0	0	0	1	0	0	0	1
0	0	0	0	1	2	0	0	2
0	0	0	0	1	2	3	0	3
0	0	0	0	1	2	3	4	4
0	0	0	0	5	2	3	4	5
6	0	0	0	5	2	3	4	6
6	7	0	0	5	2	3	4	7
6	7	0	8	5	2	3	4	8
6	7	9	8	5	2	3	4	9
10	7	9	8	5	2	3	4	10
Etc.								

3. Montrer les propriétés de vivacité et de sûreté

- Sûreté : Dans les algorithmes à jeton, le seul moyen de rentrer en section critique est de posséder l'unique jeton. Donc deux sites ne peuvent pas rentrer en section critique en même temps grâce à l'unicité du jeton.
- Vivacité : Le déplacement perpétuel du jeton à travers tous les sites. Ce qui assure que le site qui se trouve dans l'état entrée finira par passer dans l'état de section critique.

Ecrire un algorithme

Soit une grille torique $n \times m$, **(avec m pair)** comme représentée Fig.2. L'objectif de l'exercice est d'écrire un algorithme d'exclusion mutuelle sur cette grille.

Le principe de cet algorithme est, en utilisant un jeton, que chaque site (i, j) transmette vers un voisin déterminé (le suivant), pour réaliser une circulation de jeton. Les sites connaissent leur identitifiants, ceux de leurs voisins et les dimensions de la grille n et m. On distinguera un site qui sera l'initiateur de l'algorithme (le premier site à envoyer un jeton).

Un site (i, j) determinera son suivant de la manière suivante :

— Si $i \neq 1$ et $i \neq n$,
  — Si j est impair, le suivant est le voisin du bas.
  — Si j est pair, le suivant est le voisin du haut.
— Si $i = 1$,
  — Si j est impair, le suivant est le voisin du bas.
  — Si j est pair, le suivant est le voisin de droite.
— Si $i = n$,
  — Si j est impair, le suivant est le voisin de droite.
  — Si j est pair, le suivant est le voisin du haut.

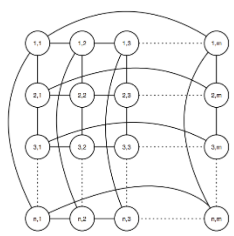


FIGURE 2 – Système distribué structuré en grille

1. Ecrire la fonction suivant() qui retournera les coordonnées du suivant (x,y) du site(i,j)

Variables :

```
n entier correspondant à la hauteur de la grille
m entier pair correspondant à la largeur de la grille
(i,j) les coordonnées du site où on se trouve
```

Fonction, qui retourne un couple de coordonnées :

```
suivant()
      si i==1 alors
            si j%2==0 alors
```

```
                retourner(i,j+1)
            sinon
                retourner(i+1,j)
    sinon
            si i==n alors
                si j%2==0 alors
                    retourner(i,j+1)
                sinon
                    retourner(i-1,j)
            sinon
                si j%2==0 alors
                    retourner(i-1,j)
                sinon
                    retourner(i+1,j)
                fin si
            fin si
    fin si

    retourner_erreur()
    sortir
fin suivant()
```

2. Ecrire le code de l'algorithme d'exclusion mutuelle pour un site (i,j)

Variables :

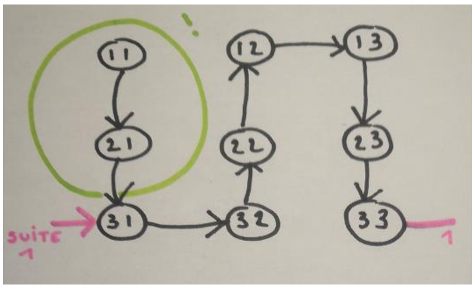
```
etat = {demande, SC, sortie}
(x,y) qui est un couple de coordonnées, avec x dans [0,n] et y dans [0,m]
```

Procédure :

```
reception_jeton()
    si etat==demande alors
        etat = SC
        ##section critique##
        etat = sortie
    fin si
    (x,y) = suivant()
    envoyer_jeton((x,y))
fin reception_jeton()
```

3. Expliquer le problème si m n'est pas pair

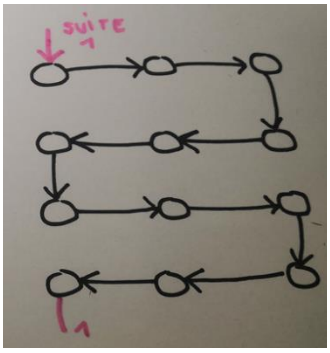
Il est important de noter que la grille est **torique**. Si m n'est pas pair on va exclure toute une partie de la grille lorsque l'on atteint le site (n,m) (partie verte).



Exemple grille torique (3,3)

4. Sans écrire l'algorithme, imaginer et décrire (avec des schémas) une solution si m est impair et n est pair

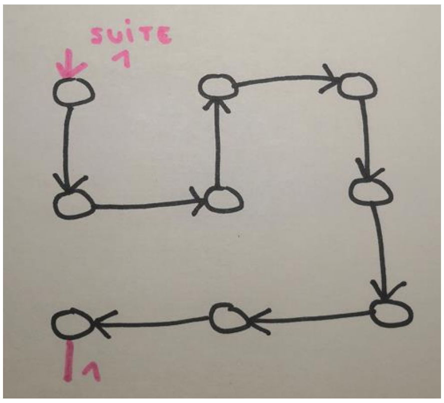
On suit l'algorithme mais de manière transposée.



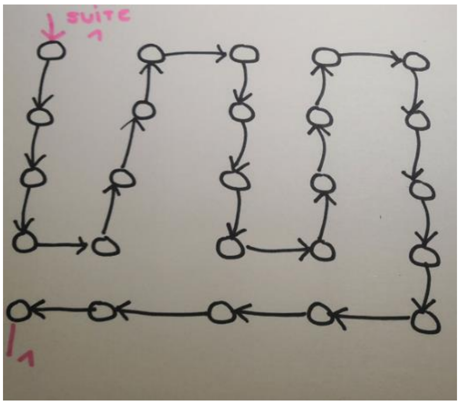
Exemple grille torique (4,3)

5. Sans écrire l'algorithme, imaginer et décrire (avec des schémas) une solution si m est impair et n sont impairs

Tant qu'on ne se trouve pas sur la dernière colonne on tourne à droite à la ligne n-1.



Exemple grille torique (3,3)



Exemple grille torique (5,5)