

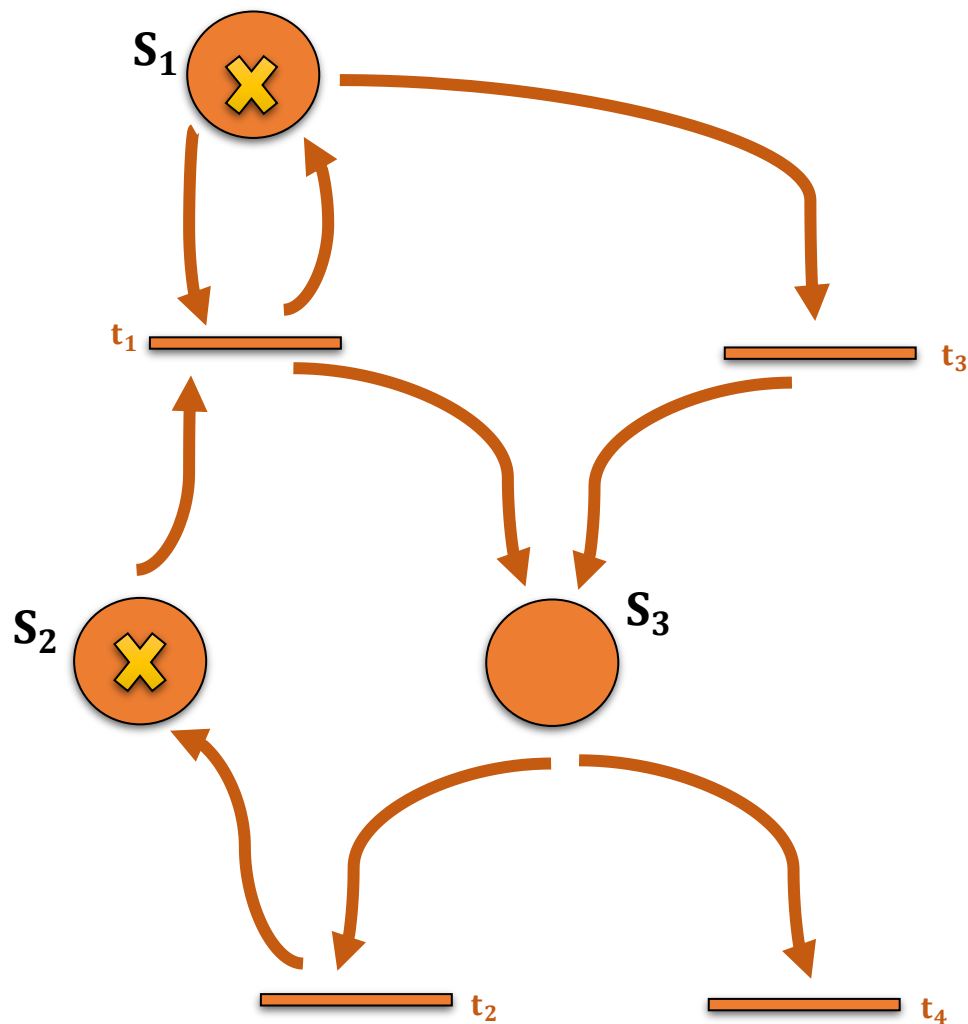
TD4

Réseaux de Petri (3) *Réseau borné, blocage, marquage accessible*

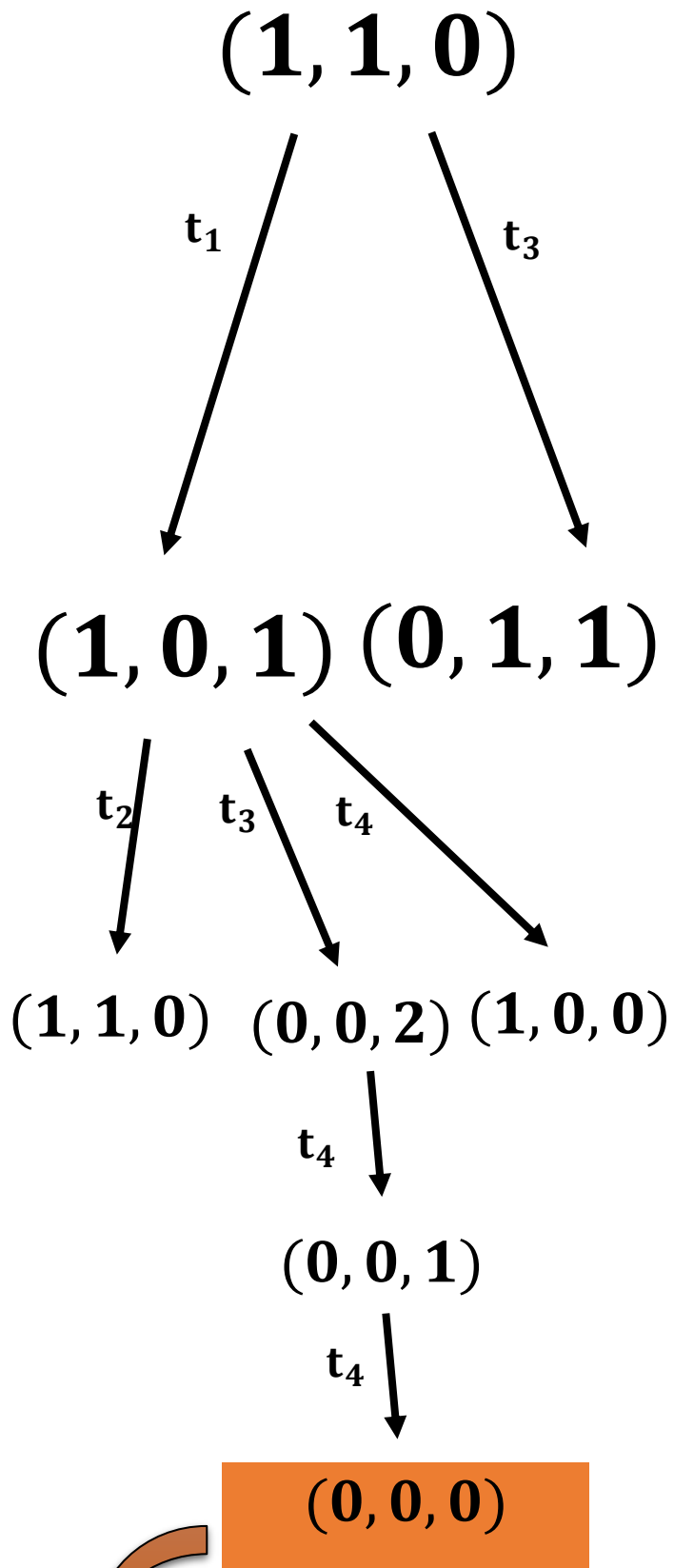
Lundi 11-10-2021

Exercice 1 • Marquage accessible

$(0, 0, 0)$ est accessible ?

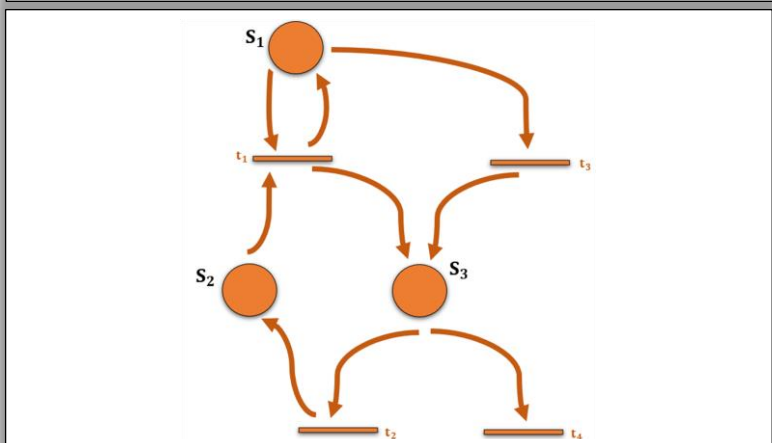
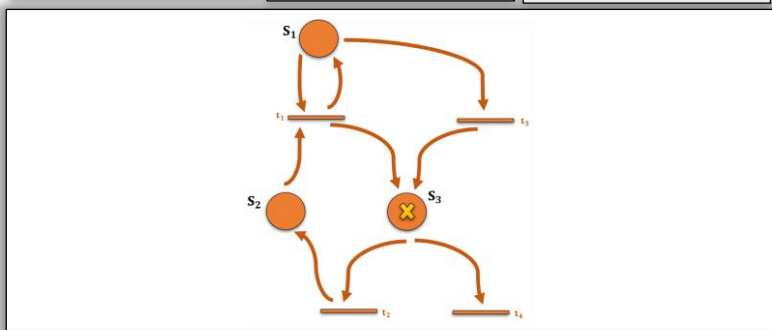
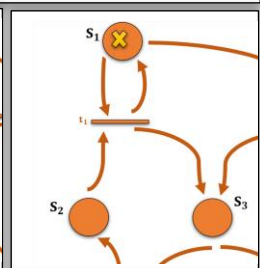
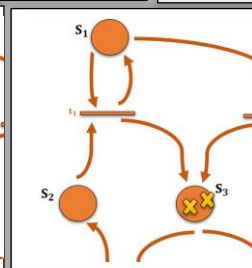
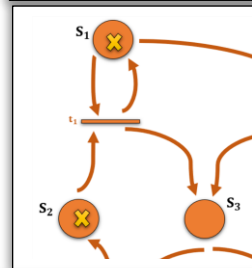
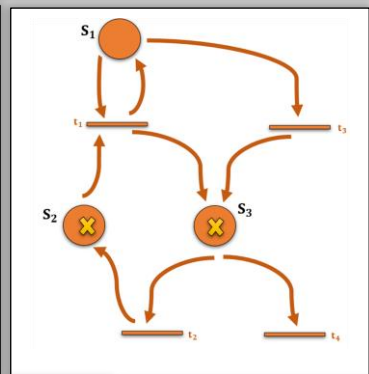
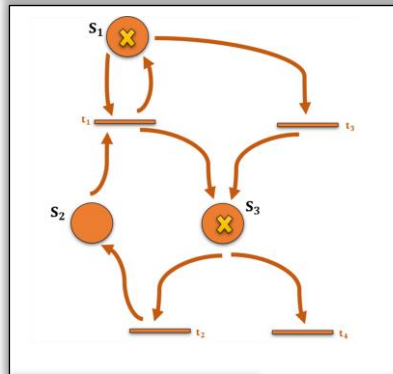
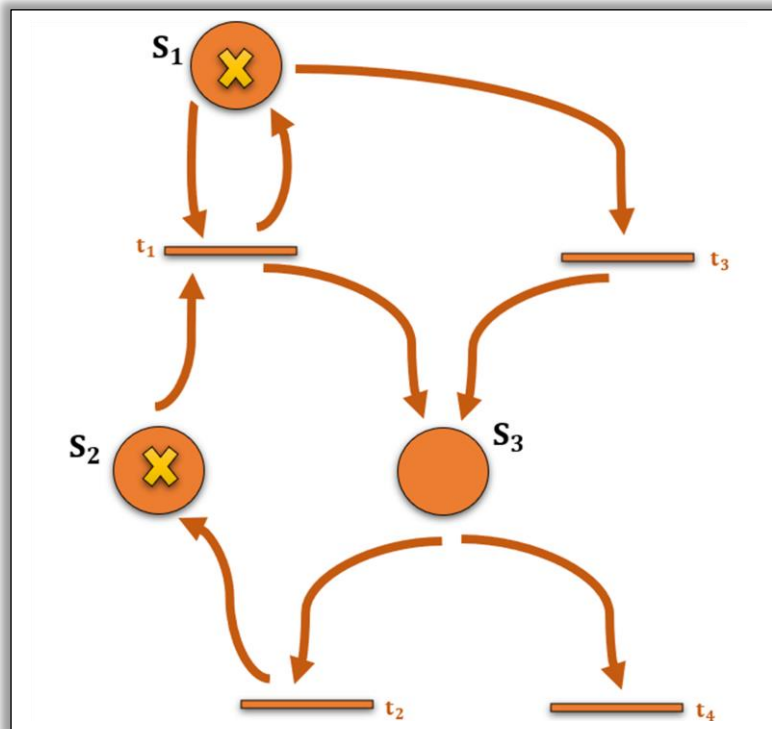


On va faire un "profondeur d'abord"



Réponse :

$(0, 0, 0)$ est accessible



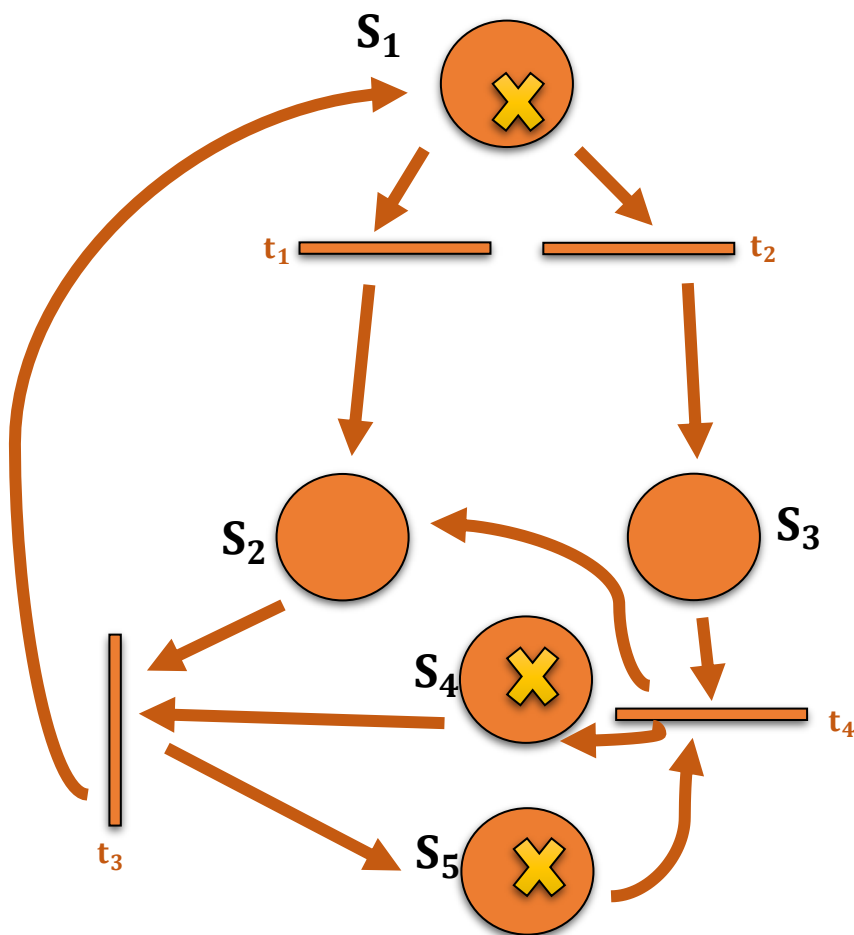
Exercice 2 . Réseau borné, blocage

Contenir un blocage ne veut pas dire que c'est borné !

Définition : Blocage

$$\exists M \quad M_0 \triangleright^* M \quad \text{et} \quad M \not\triangleright$$

Réseau borné ? Il y a un blocage?



(1, 0, 0, 1, 1)

t_1

t_2

(0, 1, 0, 1, 1)

(0, 0, 1, 1, 1)

t_3

t_4

(1, 0, 0, 0, 2)

(0, 1, 0, 2, 0)

t_1

t_2

t_3

(0, 1, 0, 0, 2) **(0, 0, 1, 0, 2)**

(1, 0, 0, 1, 1)

Blocage !

Car on ne peut rien
exécuter, il n'y a pas
d'issus

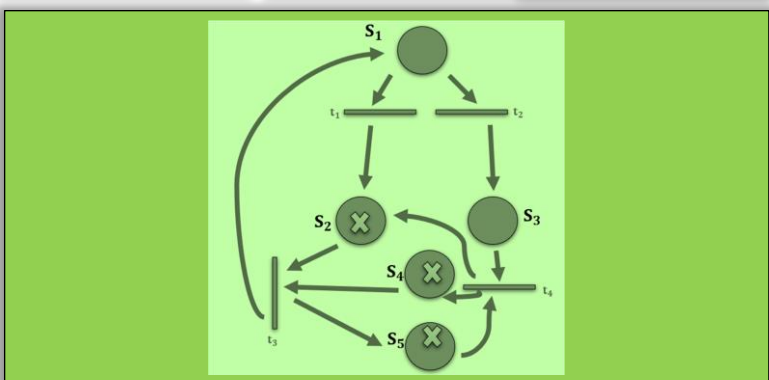
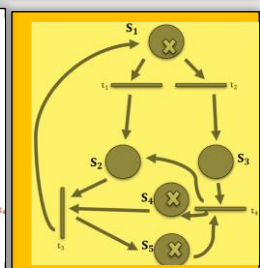
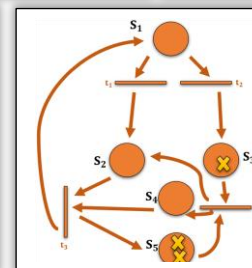
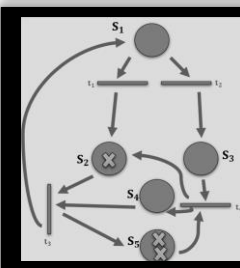
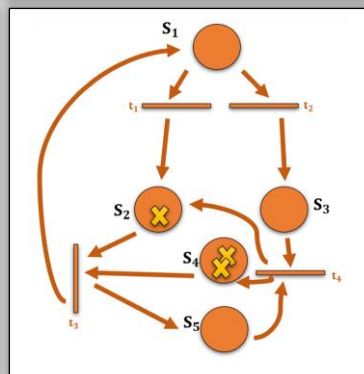
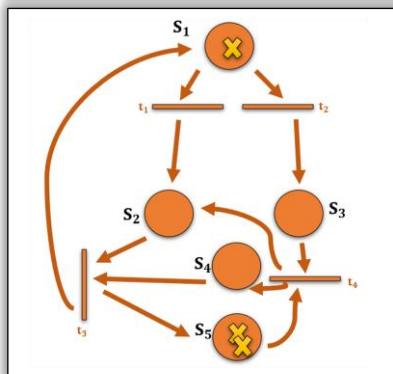
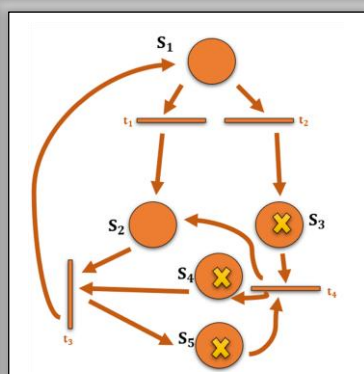
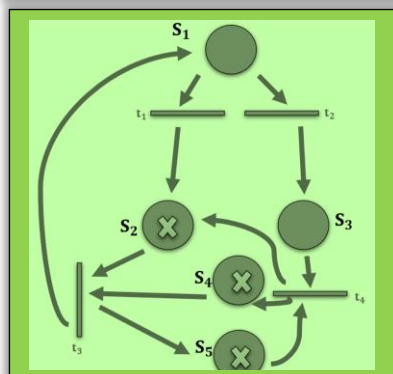
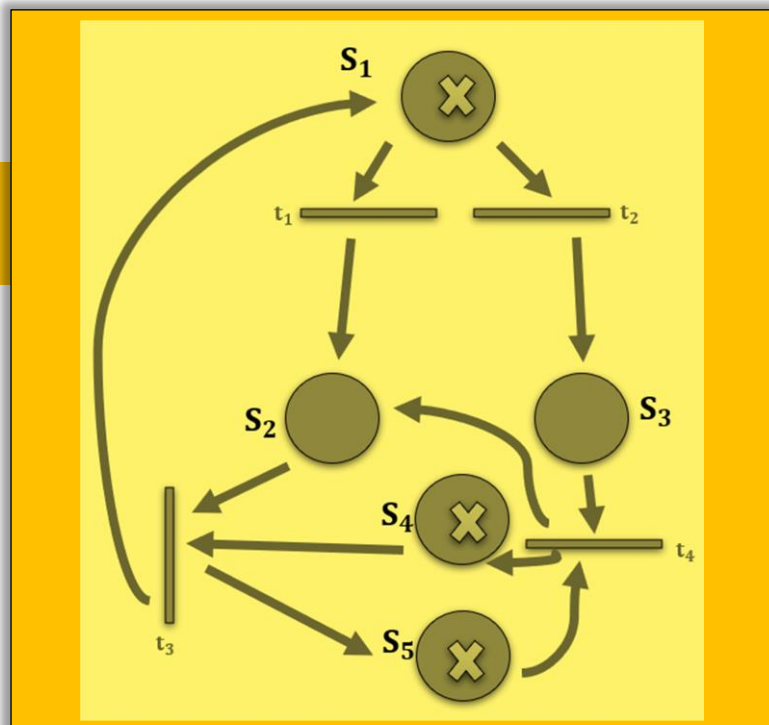
t_4

(0, 1, 0, 1, 1)

On a déjà vu ce
sommet, donc on
s'arrête.

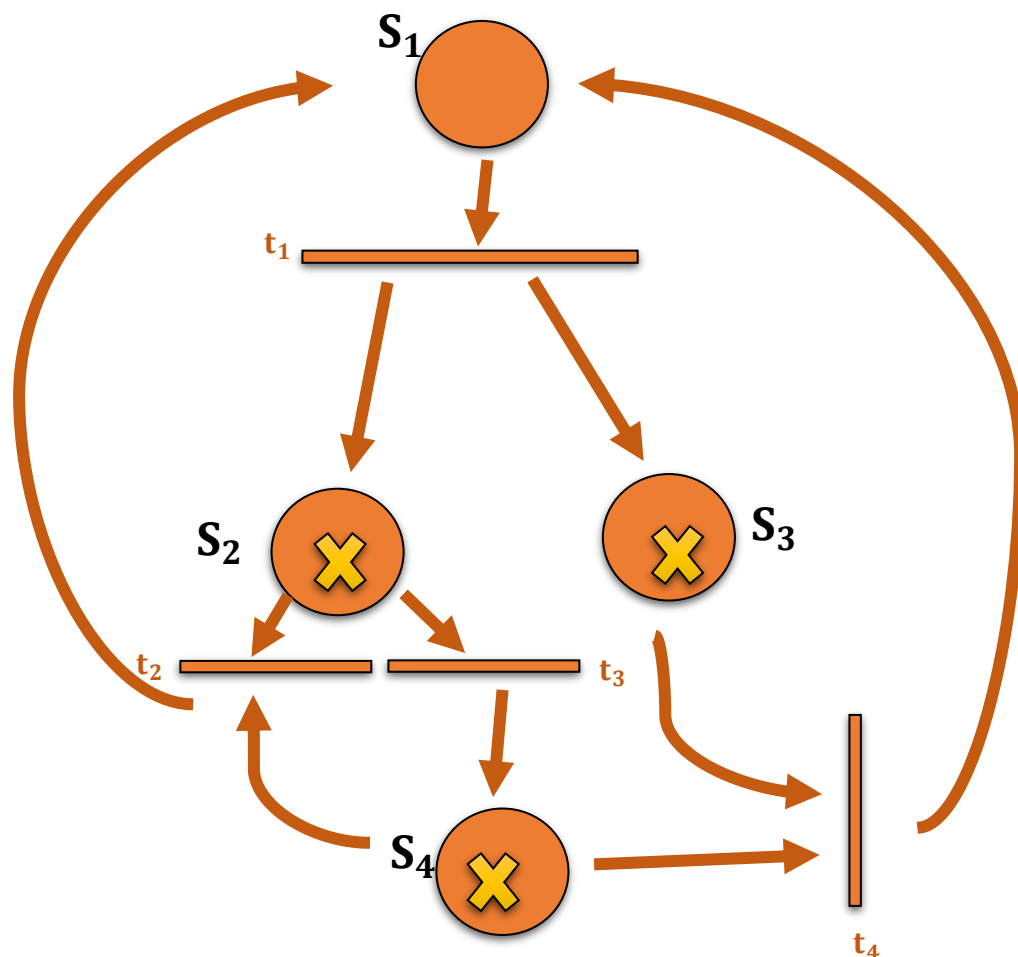
Réponse :

Il y a un blocage, il est borné à 2



Exercice 3 • Réseau borné, blocage (2)

Réseau borné ? Il y a un blocage ?



(1, 0, 0, 1)

t_1

(0, 1, 1, 1)

t_2

t_3

t_4

(1, 0, 1, 0)

(0, 0, 1, 2)

(1, 1, 0, 0)

t_1

t_4

t_1

t_3

(0, 1, 2, 0)

(1, 0, 0, 1)

(0, 2, 1, 0)

(1, 0, 0, 1)

t_3

*Déjà vu,
donc on
s'arrête*

t_3

*Déjà vu,
donc on
s'arrête*

(0, 0, 2, 1)

(0, 1, 1, 1)

*Déjà vu,
donc on
s'arrête*

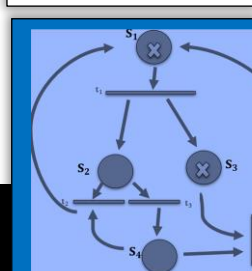
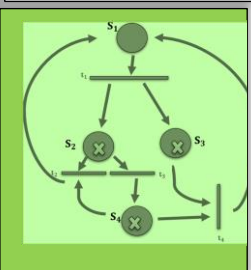
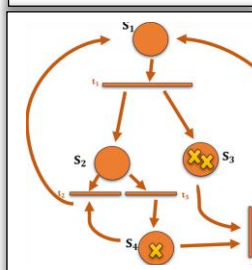
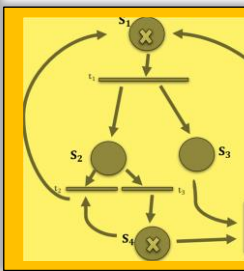
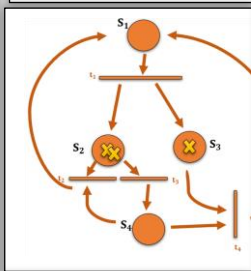
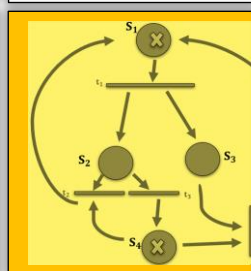
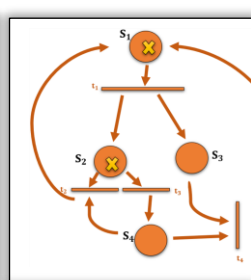
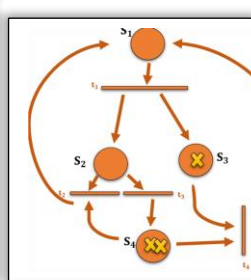
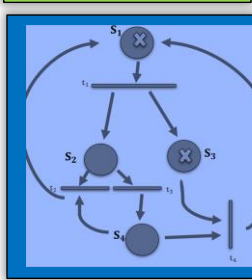
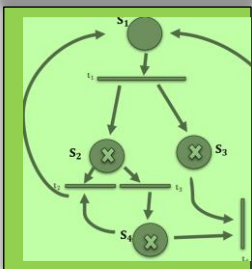
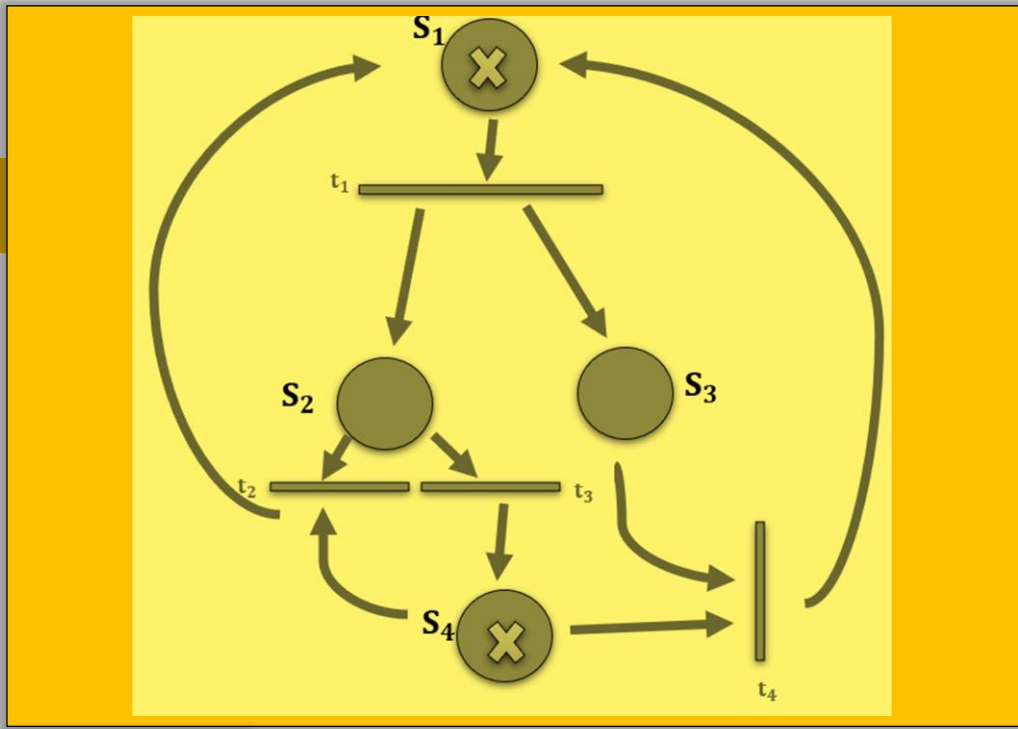
t_4

(1, 0, 1, 0)

*Déjà vu,
donc on
s'arrête*

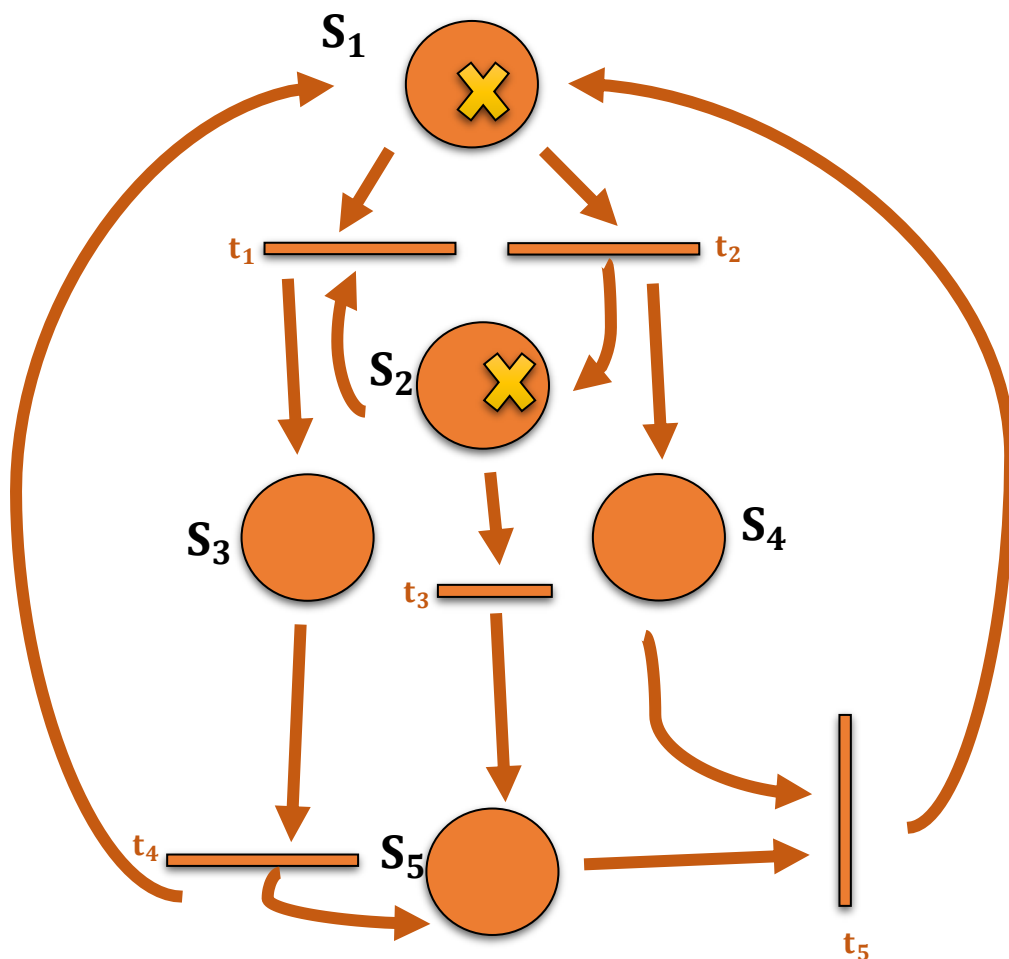
Réponse :

pas de blocage, il est borné



Exercice 4 • Réseau borné, blocage (3)

Réseau borné ? Il y a un blocage ?



On va faire un "profondeur d'abord"

(1, 1, 0, 0, 0)

t_1

t_2

t_3

(0, 0, 1, 0, 0)

(0, 2, 0, 1, 0)

(1, 0, 0, 0, 1)

Déjà vu

t_4

t_3

(1, 0, 0, 0, 1)

(0, 1, 0, 1, 1)

Déjà vu

t_2

(0, 1, 0, 1, 1)

t_3

t_5

(0, 0, 0, 1, 2)

(1, 1, 0, 0, 0)

Déjà vu

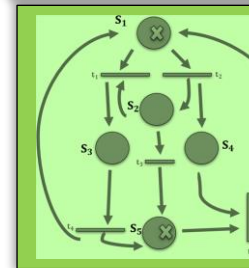
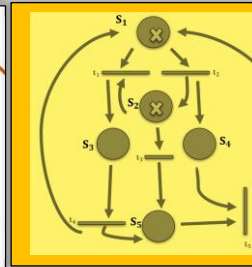
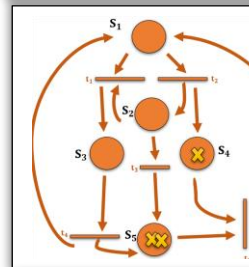
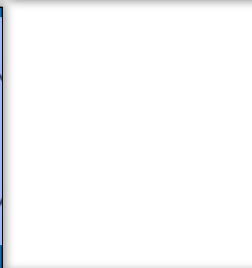
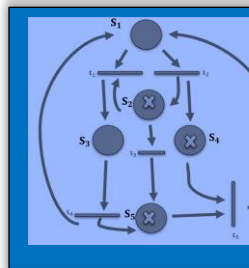
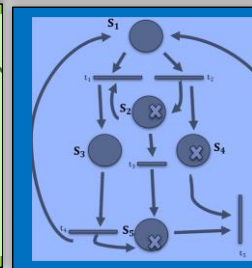
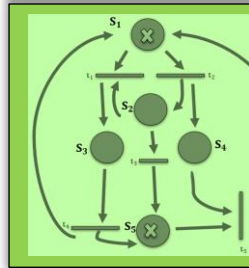
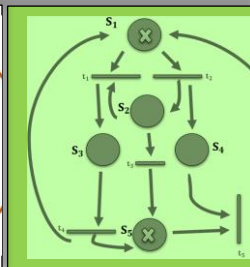
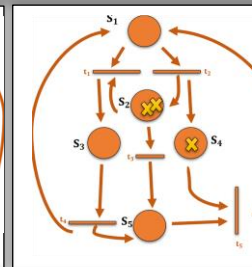
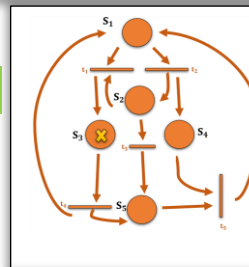
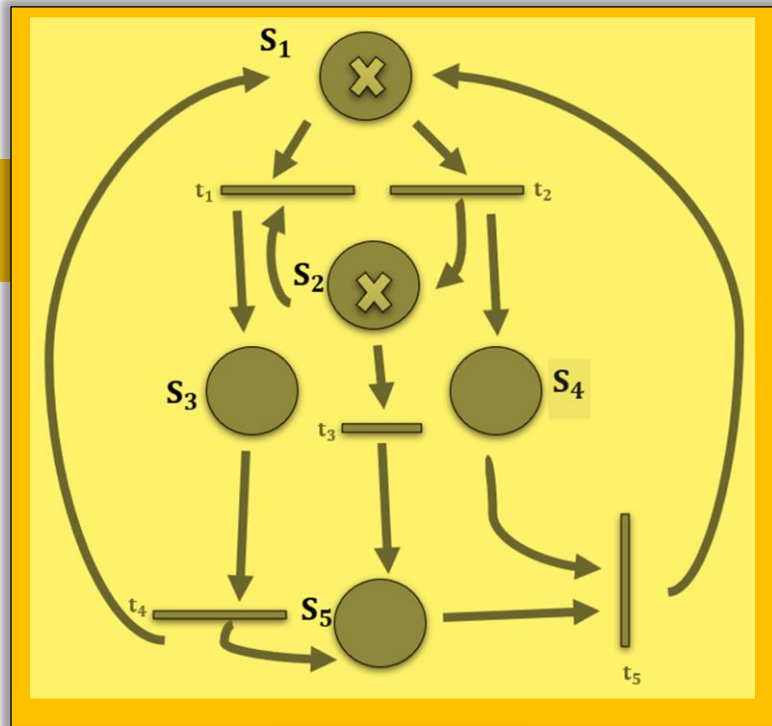
t_5

(1, 0, 0, 0, 1)

Déjà vu

Réponse :

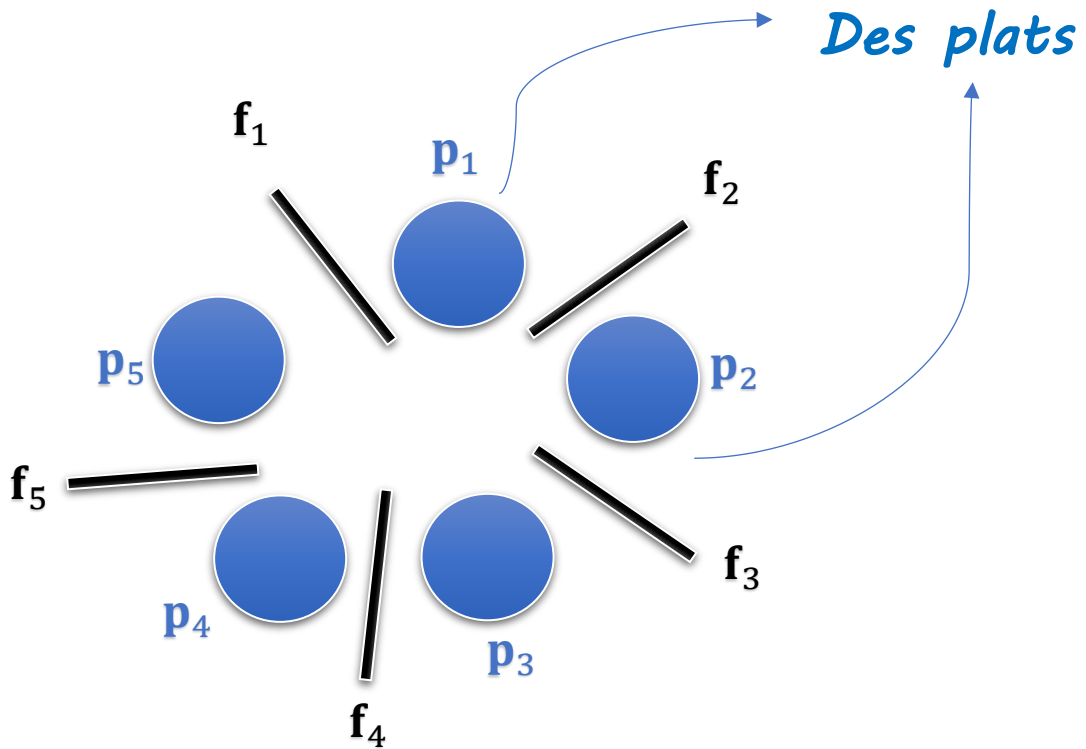
pas de blocage, il est borné



Le dîner des philosophes

Le diner des philosophes

- Les philosophes ont 2 activités : manger / penser.
Soit il sont en train de manger, soit il sont en train de penser.
- Pour manger, chaque personne a besoin de 2 outils :
2 baguettes / 2 fourchettes
- Ces outils sont partager avec les voisins

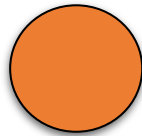


Ecrire un modele pour un pilosof p_i et le relier avec les ressources.

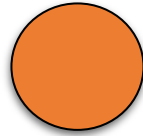
Un pilosof a besoin de 2 fourchettes pour manger.

1 pilosof : soit il mange, soit il pensse. Ce sont les 2 etats essensiel : **refleshir, manger**

Manger



Refleshir



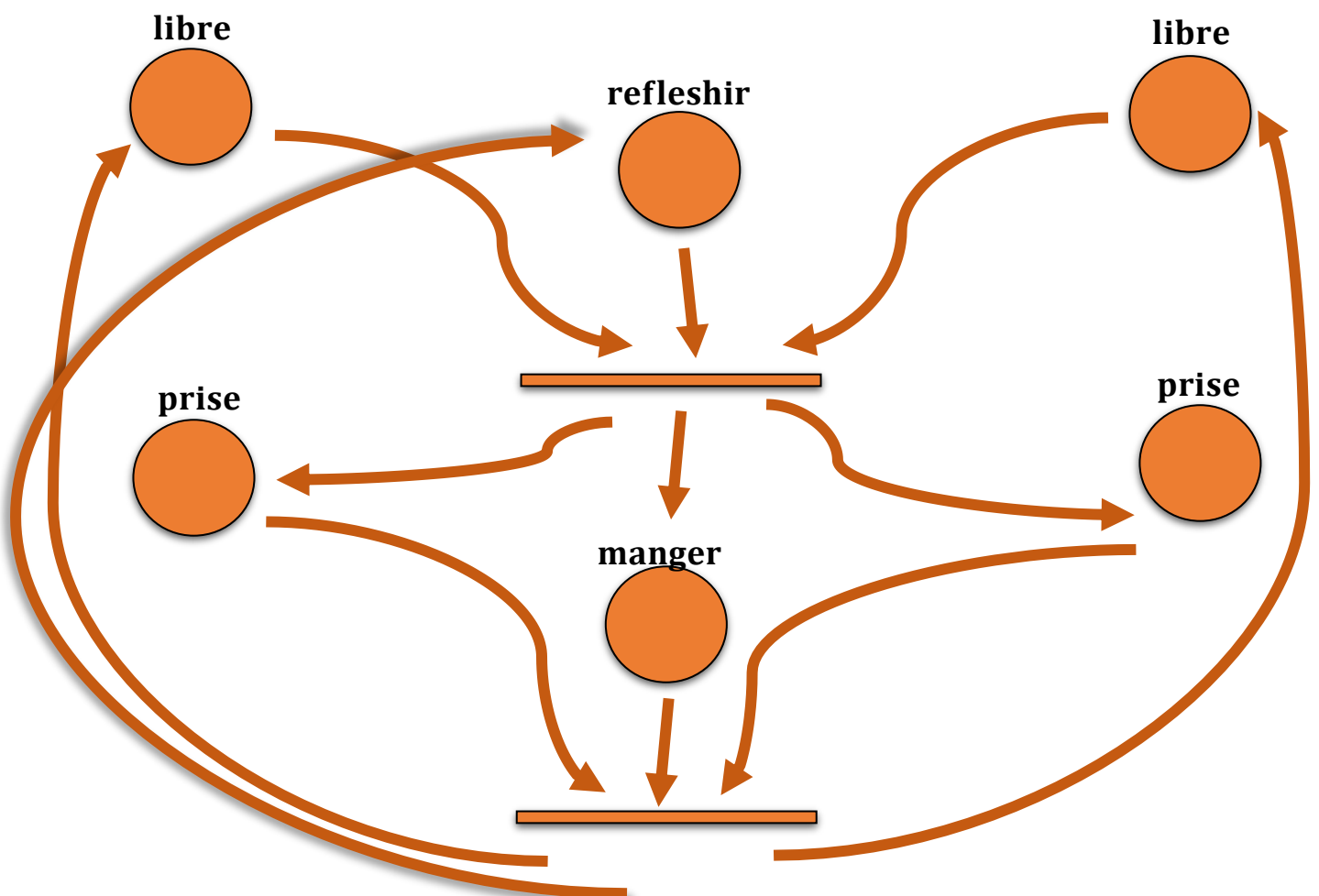
Maintenant, comment passer d'un etat a l'autre ?

C'est quoi les etats d'une baguet ? soit elle est libre, soit elle est prise.

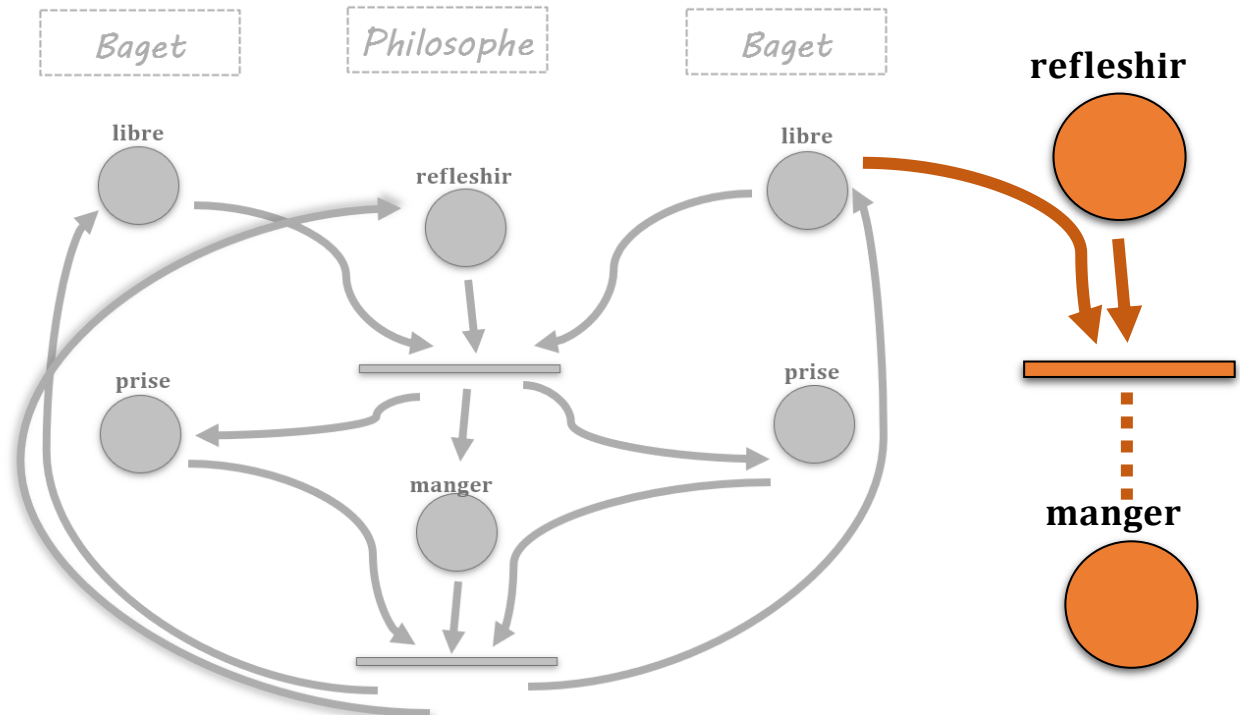
Baget

Philosophe

Baget

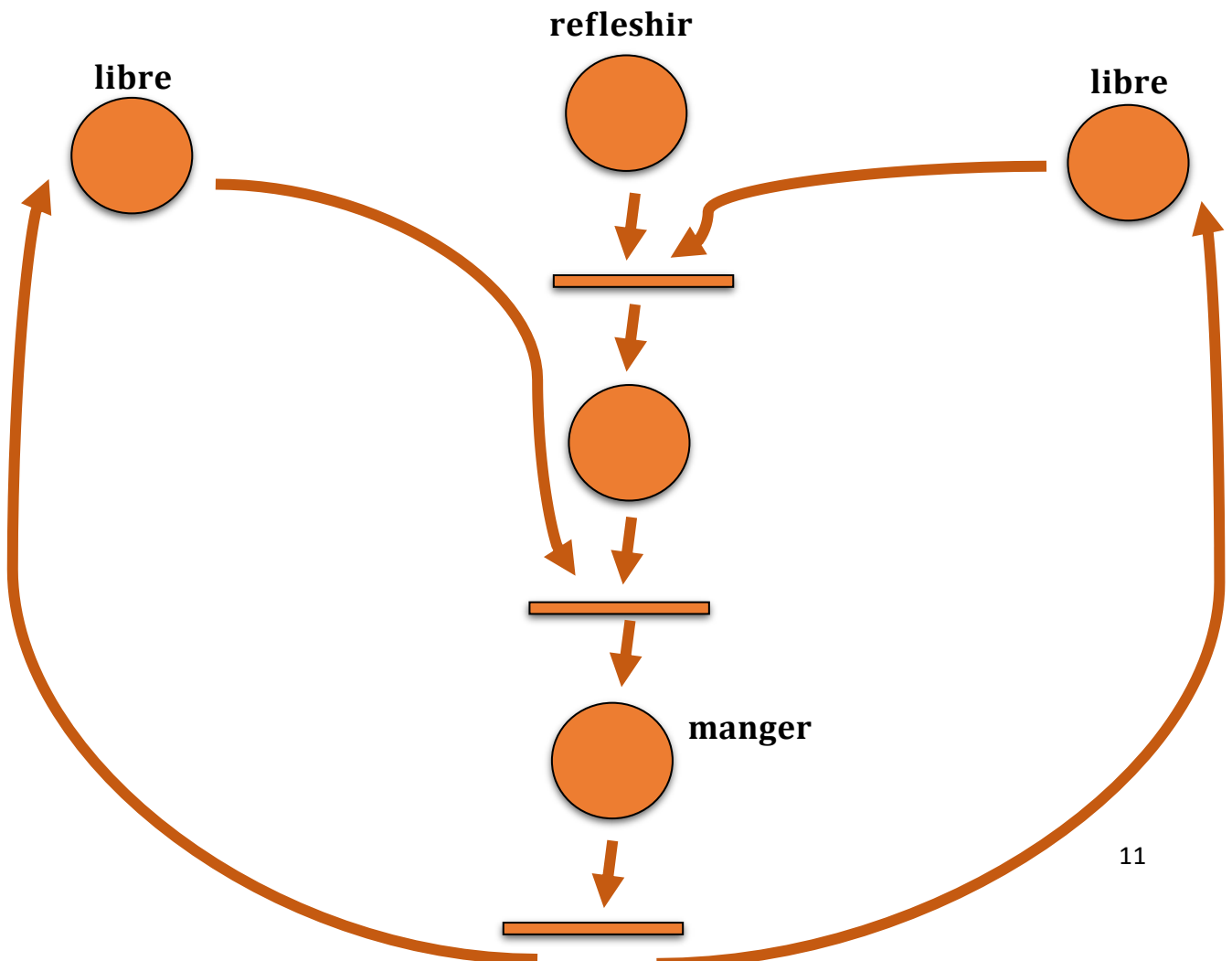


Si on ajoute un 2em philosophe



Autre solution ?

S'il y a une baget libre, je la prends, même si la 2em est pas encore libre.

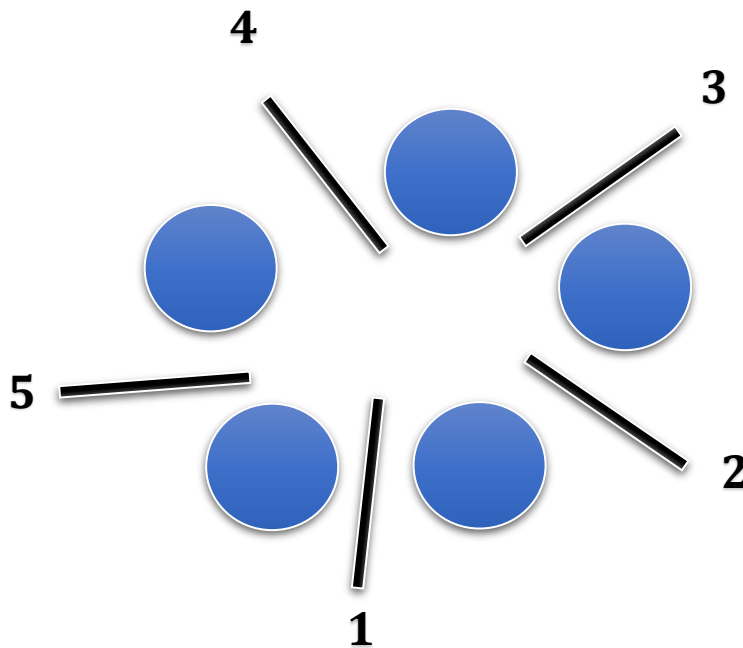


Le problème : *Si tlm prenne une baguette, on est bloqué.*

- *Situation typique d'inter-blocage dans les machines concurrentes.*
- *Est-ce que le pbm de blocage existe dans la 1^{ère} solution ? Non, car ils prennent les 2 bagets à la fois, cette solution nécessite qu'on sache que les 2 existent, ça demande une synchronisation, ce que d'habitude on essaye d'éviter car la synchronisation entre 2 ressources ralentit le système.*

Autre solution

On demande à chaque processus de prendre les bagets dans l'ordre croissant des numéros :



.