جامعة محمد الخامس السويسي
Université Mohammed V - Souissi

ENSIAS

Corrigé TD

[modélisation logicielle]

جامعة محمد الخامس السويسي
Université Mohammed V - Souissi

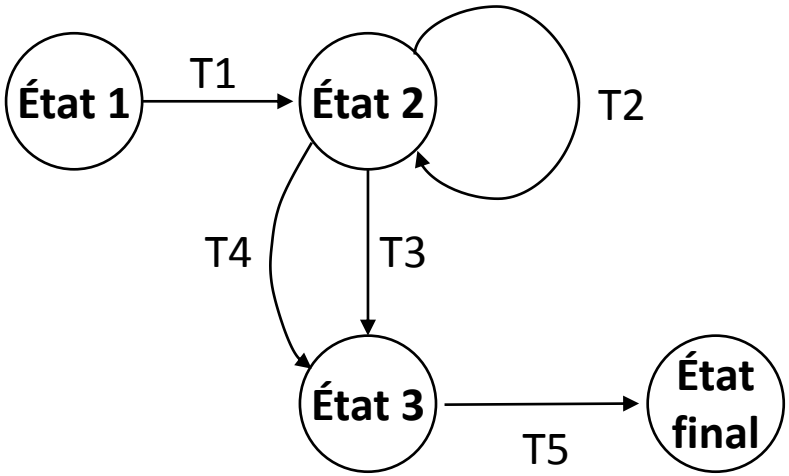
ENSIAS

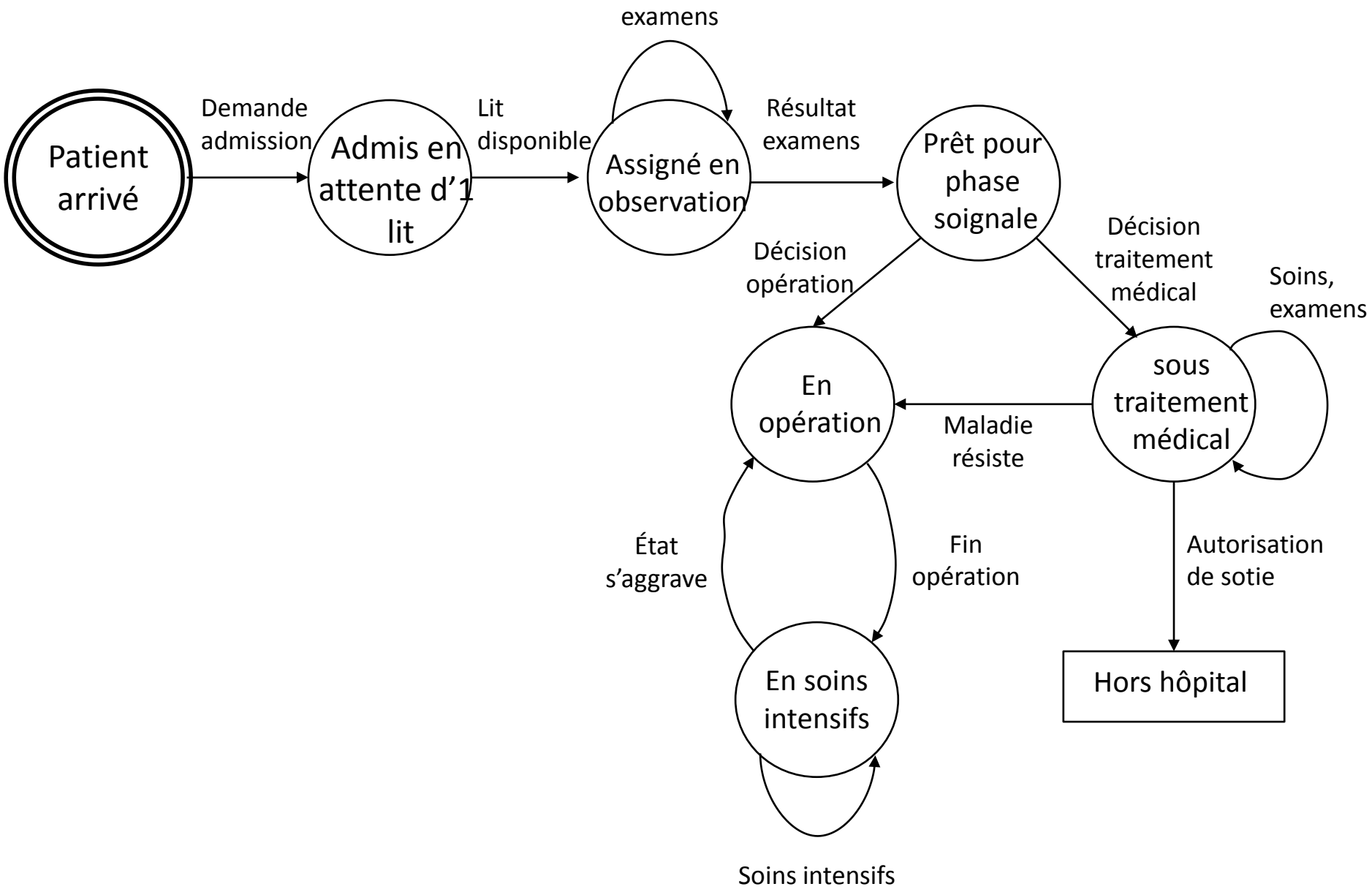
Corrigé TD1

[modélisation à l'aide des machines E/T]

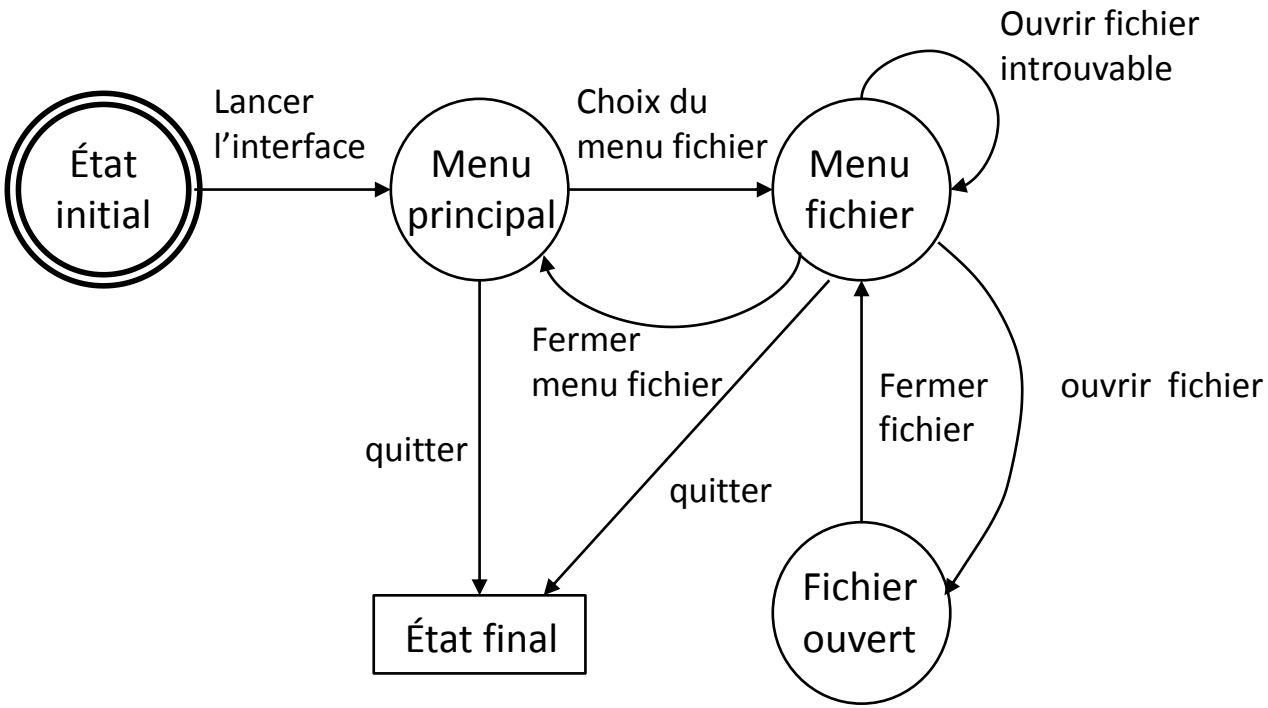
Exercice 1: automates finis

Ce type de modélisation prend la forme générale suivante




Patient arrivé
Demande admission
Admis en attente d'1 lit
Lit disponible
Assigné en observation
examens
Résultat examens
Prêt pour phase soignale
Décision opération
Décision traitement médical
Soins, examens
En opération
sous traitement médical
Maladie résiste
État s'aggrave
Fin opération
Autorisation de sotie
En soins intensifs
Hors hôpital
Soins intensifs

Exercice 2: interface



Exercice 3:

événements / états	NV Etats T1 action	AP Etat T2 action	AU Etat T3 action	FV Etat T4 action
E0	E1 A1			
E1		E1 A2	E1 A3	Ef A4
Ef				

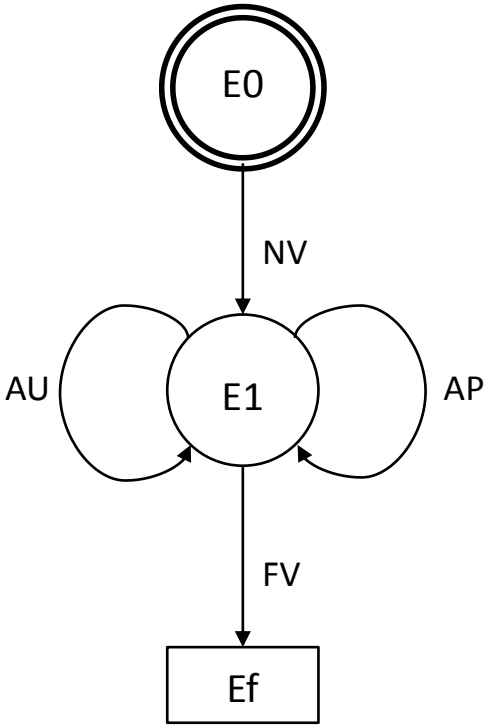
A1 :mise en route de la caisse
mise à 0 totaliseur
impression logo et date

A2 :recherche nom et prise du
poids unitaire dans la table des prix
calcul prix de la quantité
impression nom, poids, prix
ajout prix au totaliseur

A3 :trouver nom et prix unitaire dans
la table des prix
calculer prix du nombre d'unités
impression nom, prix, nombre
d'unités

A2 :imprimer contenu totaliseur
Arrêter la caisse

Diagramme E/T:



Légende:

NV	:nouvelle vente
FV	:fin vente
AU	:article à l'unité
AP	:article au poids

جامعة محمد الخامس السويسي
Université Mohammed V - Souissi

ENSIAS

Corrigé TD2

[modélisation à l'aide des réseaux de Pétri]

énoncé

<u>Exercice 1:</u> modélisation du diner des philosophes avec les réseaux de Pétri

4 philosophes sont assis autour d'une table et on dispose de 4 baguettes. Chaque philosophe a besoin de deux baguettes pour manger. Donc seuls les philosophes qui se trouvent l'un en face de l'autre peuvent manger en même temps, alors que les philosophes adjacents ne le peuvent pas.

Ce problème mais en jeu des ressources partagées (les baguettes), d'où le choix de modélisation avec les RdP.

<u>Exercice 2</u>: modélisation de l'accès de 2 calculateurs à une mémoire.

2 calculateurs ont accès à une mémoire commune. On suppose que chaque calculateur peut se trouver dans l'un des états suivants:

- Le calculateur s'exécute sans avoir besoin de la mémoire
- Le calculateur demande l'accès à la mémoire
- Le calculateur utilise la mémoire

Modéliser ce fonctionnement à l'aide des RdP

NB: la mémoire est une ressource partagée qui est soit utilisée soit libre.

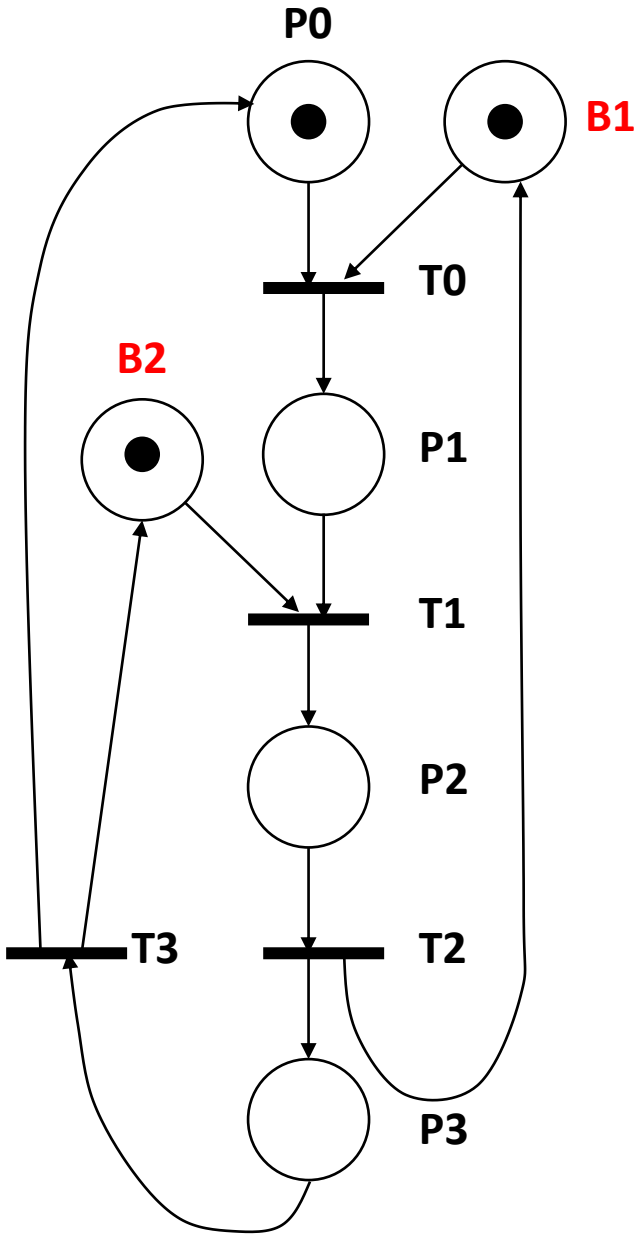
<u>Exercice 3</u>: transformation d'un RdP à capacité en un RdP ordinaire.
(voir corrigé)

Exercice 1:

- **Solution 1**:

Une première solution consiste à ce que chaque philosophe soit d'abord dans un état de pensée, puis prendre la baguette de droite, puis celle de gauche, puis manger, et après poser la baguette de droite et enfin celle de gauche avant de retourner à l'état de pensée.

Cette situation est représentée par le RdP suivant, présentant 1 seul philosophe.



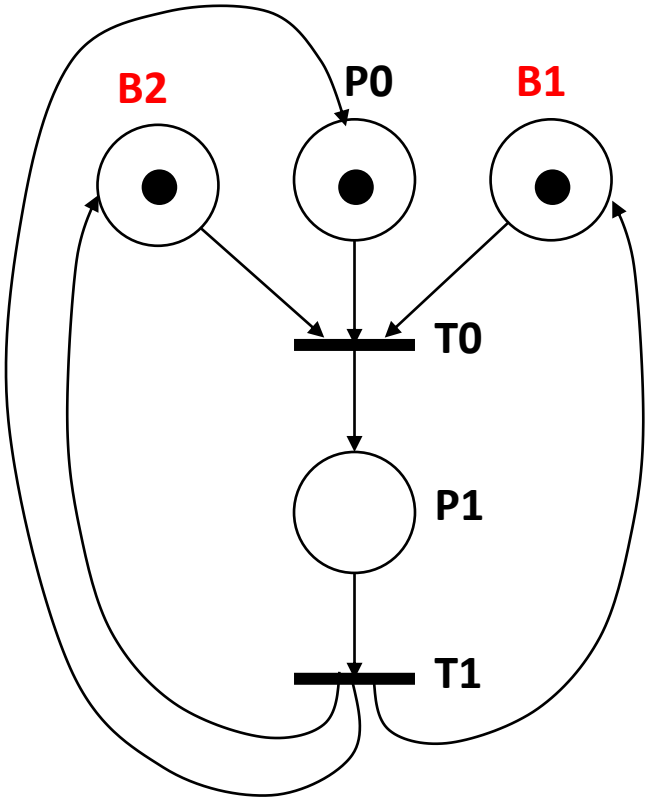
- **Légende**:

P0: le philosophe pense
B1, B2: les baguettes
P1: le philosophe a la baguette droite
P2: le philosophe peut manger
P3: le philosophe mange
T2: le philosophe pose la baguette droite
T3: le philosophe pose la baguette gauche et retourne à son état initial.

Cette solution présente un risque de blocage (deadlock) si tous les philosophes prennent en même temps une baguette.

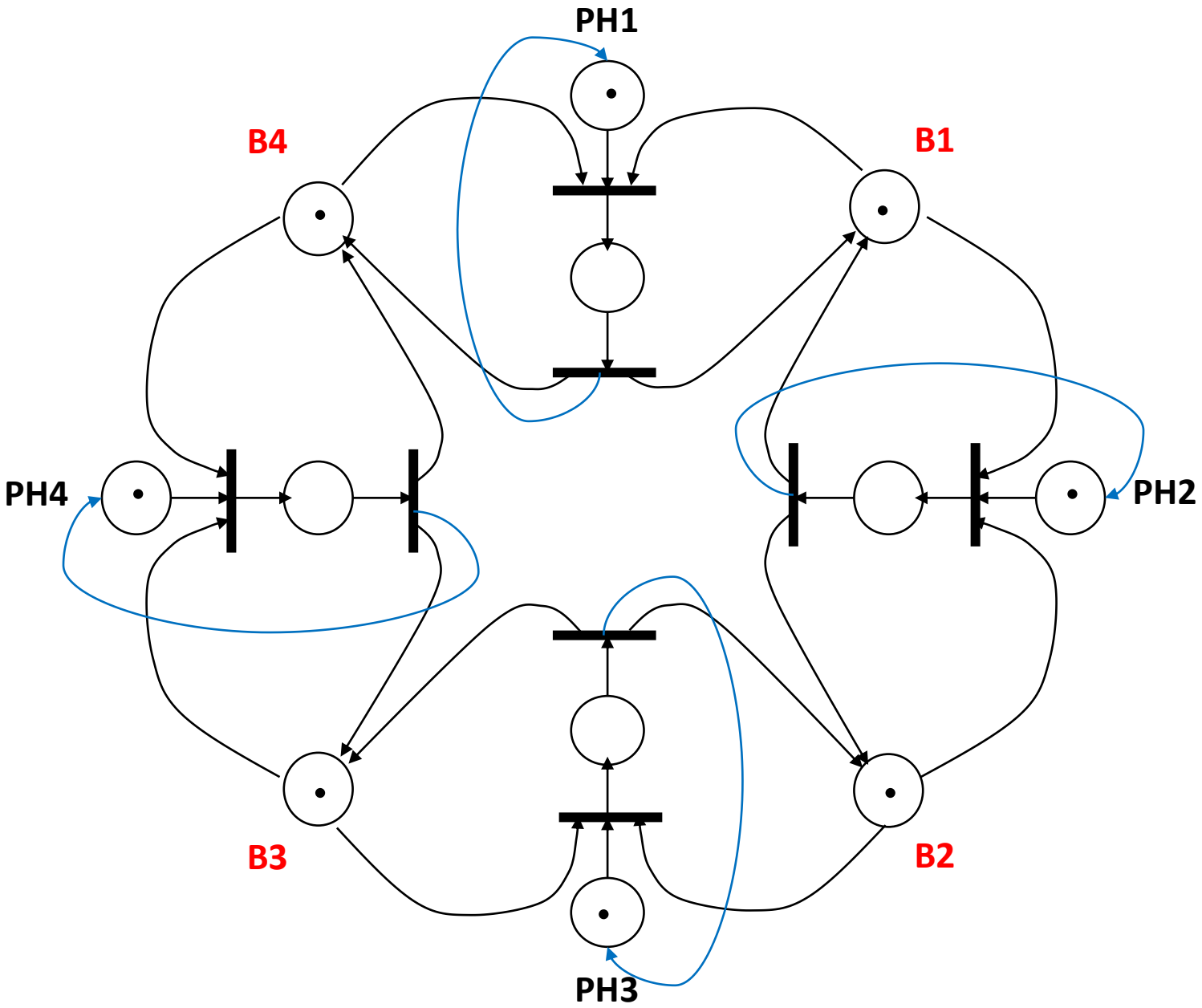
- **Solution 2**:

Maintenant on suppose que le philosophe doit disposer des deux baguettes en même temps pour manger. Cette solution est modélisée par le RdP ci-contre.

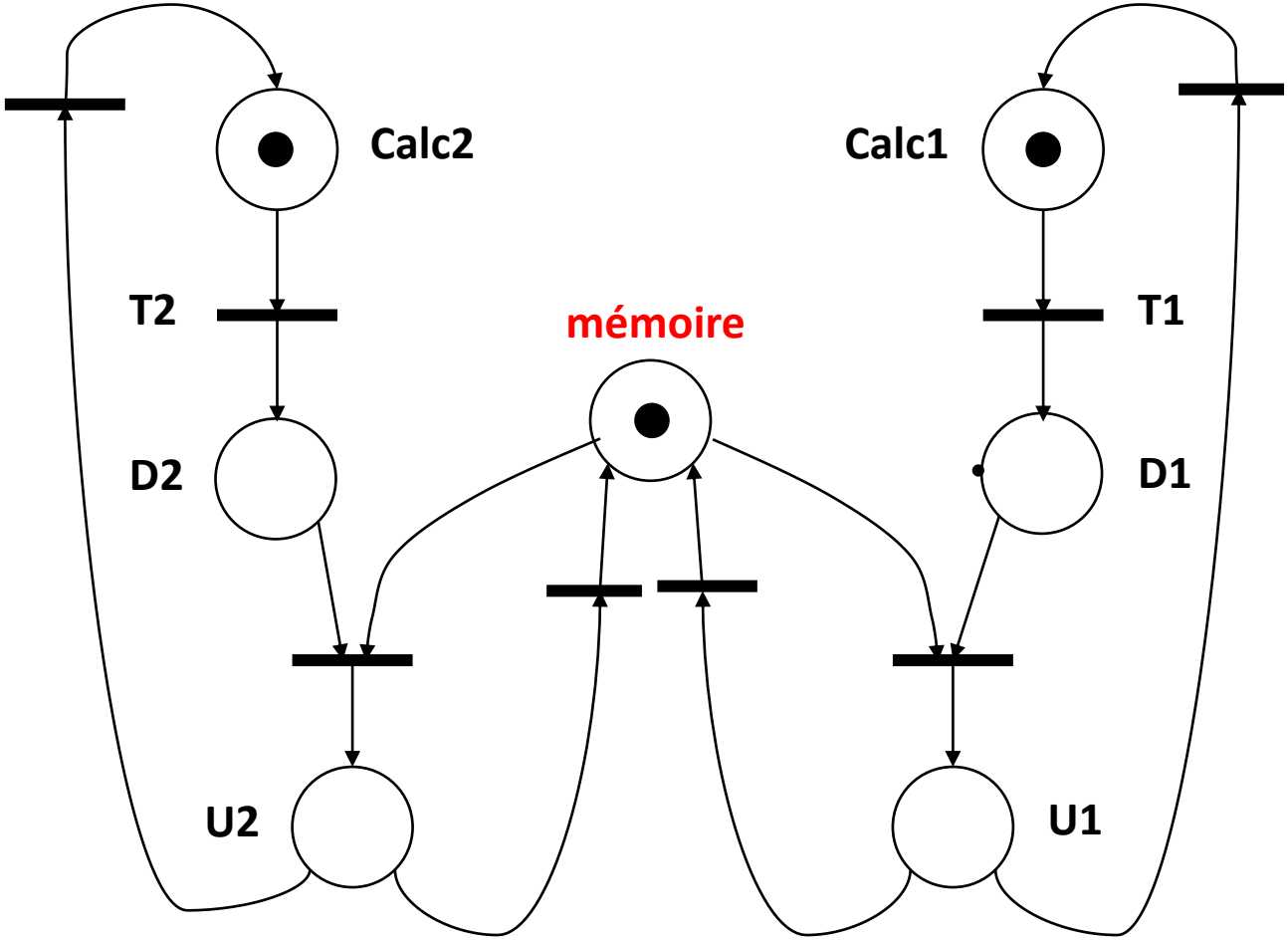


Cette solution représente un seul philosophe. Or les baguettes sont partagées entre les 4 philosophes. La solution suivante présente la solution finale.

- **Solution 3**:



Exercice 2:

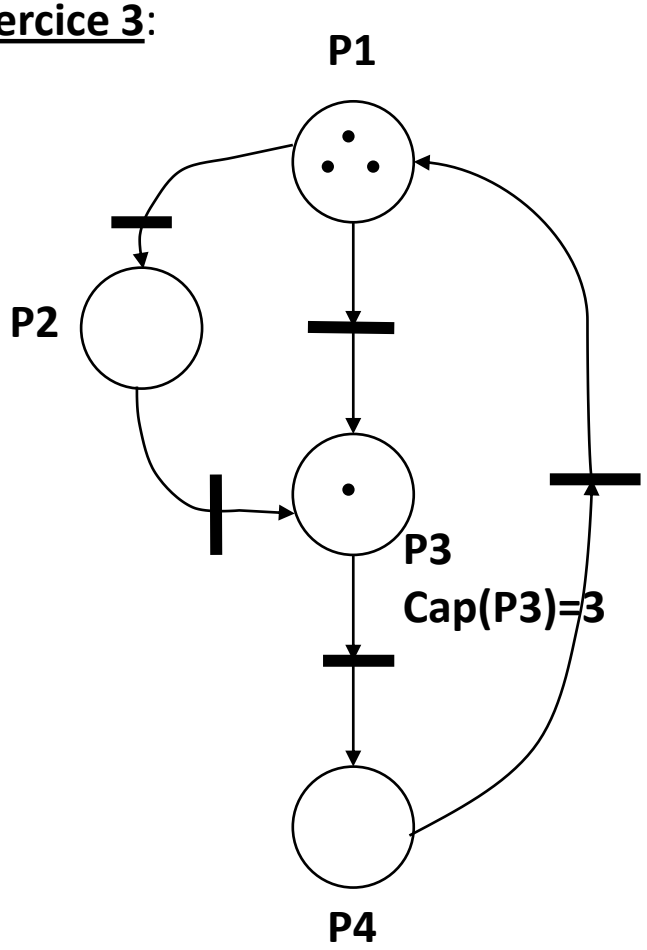


- **Légende**:

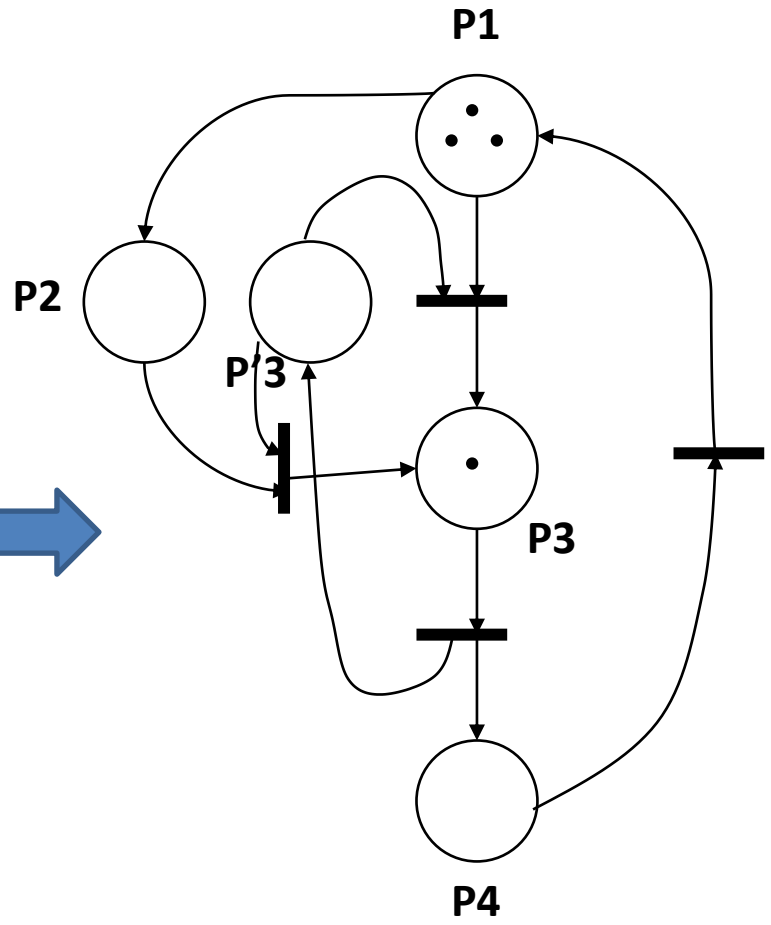
Di : le calculateur i demande la mémoire
Ui : le calculateur i utilise la mémoire
calci: : le calculateur i s'exécute et n'a pas besoin de la mémoire

NB: si l'état de la demande de la mémoire est absent, l'autre calculateur n'aura pas la chance d'utiliser la mémoire. L'état de la demande provoque un délai pendant lequel l'autre peut utiliser la mémoire.

Exercice 3:



RdP à capacité



RdP ordinaire

NB: la place P'3 et P3 dans le schéma équivalent doivent contenir en somme la capacité initiale , c'est-à-dire qu'à chaque instant, la somme des jetons dans les places P3 et P'3 est égale à 3.

جامعة محمد الخامس السويسي
Université Mohammed V - Souissi

ENSIAS

Corrigé TD3

[modélisation à l'aide des réseaux de Pétri]

énoncé

Exercice 1:

Imaginez une banque qui a deux guichets et une file d'attente pour l'accès aux guichets. La file d'attente a 2 portes, une pour chaque guichet. Lorsqu'un guichet est occupé et qu'il y a quelqu'un dans la file, c'est la porte de l'autre guichet qui s'ouvre s'il est libre et réciproquement. Si les deux guichets sont occupés, les deux guichets sont ouverts et il n'y a personne dans la file d'attente.

Modéliser le fonctionnement des guichets par un réseau de pétri.

Exercice 2:

Un bain turc contient 5 places et 7 masseurs, les clients qui arrivent sont acueuillis dans une salle d'attente. Le bain nécessite un masseur et se passe dans une salle de bain.

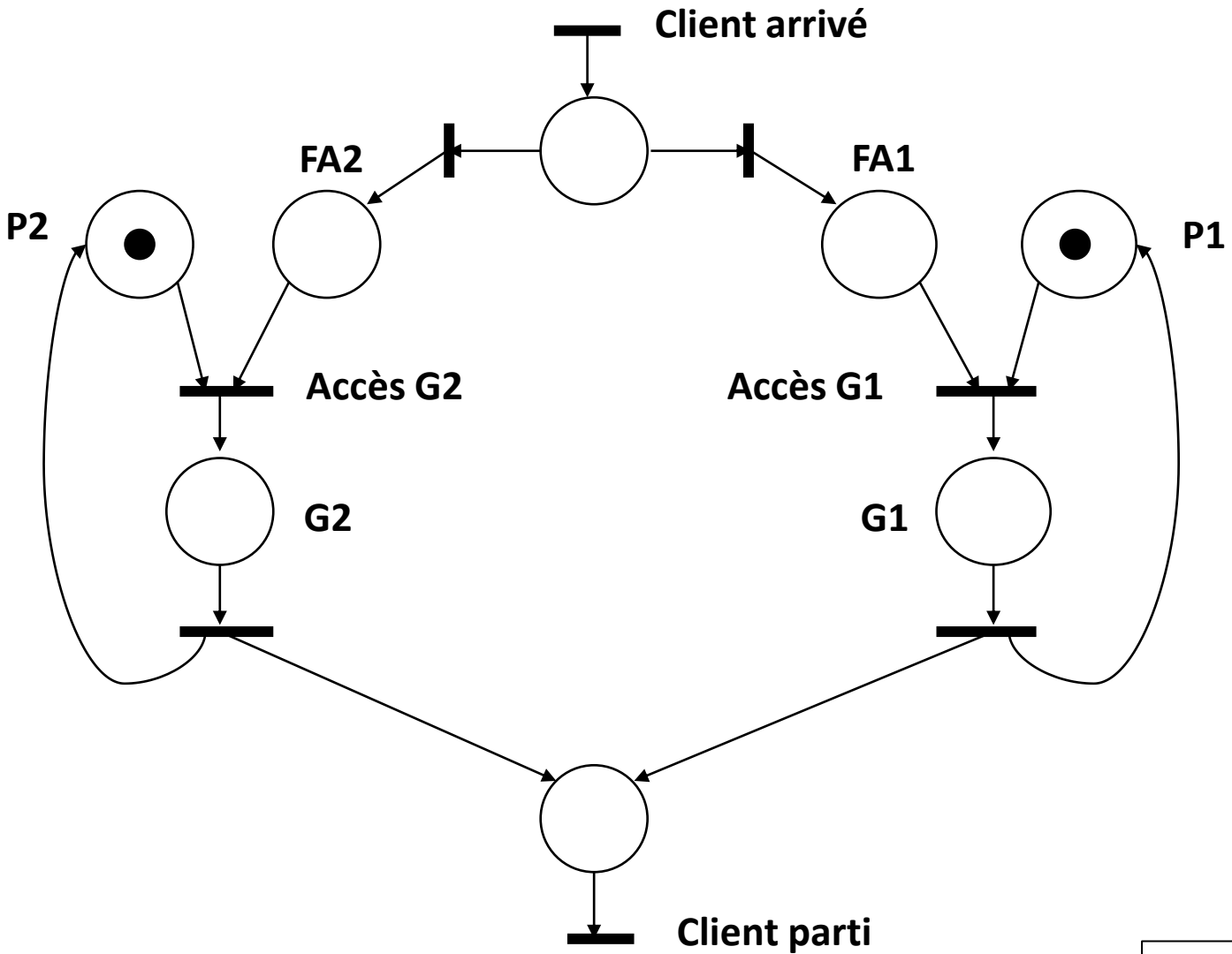
Modéliser le fonctionnement du bain turc à l'aide d'un RdP.

Exercice 3:

Un système de production reçoit en entrée des matières brutes sur lesquelles il applique des transformation pour donner en sortie des produits finis. Le système est géré par la méthode KANBAN qui consiste à avoir en un instant donné, une quantité de produits finis exactement égale à la capacité de stockage.

Modéliser le fonctionnement de ce système de production avec un RdP.

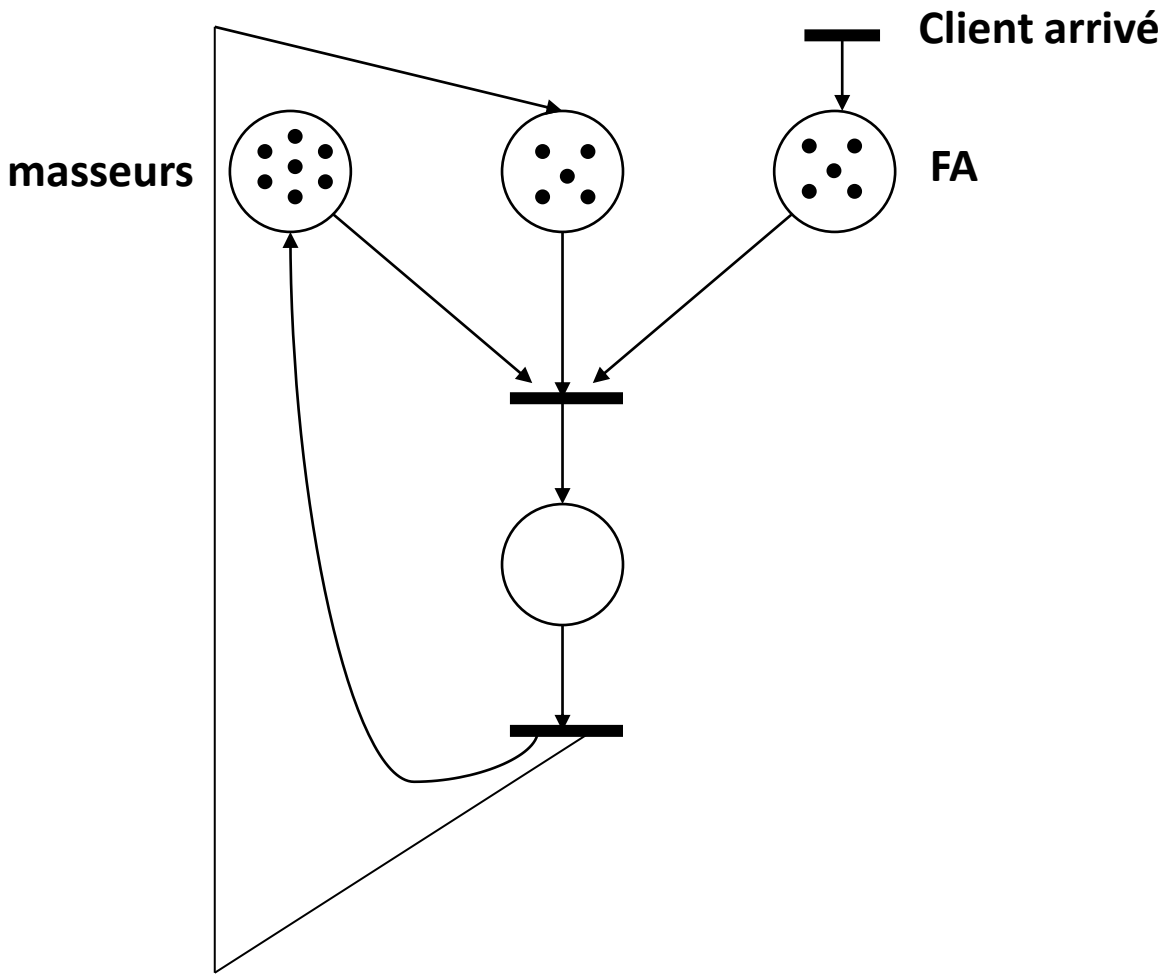
Exercice 1:



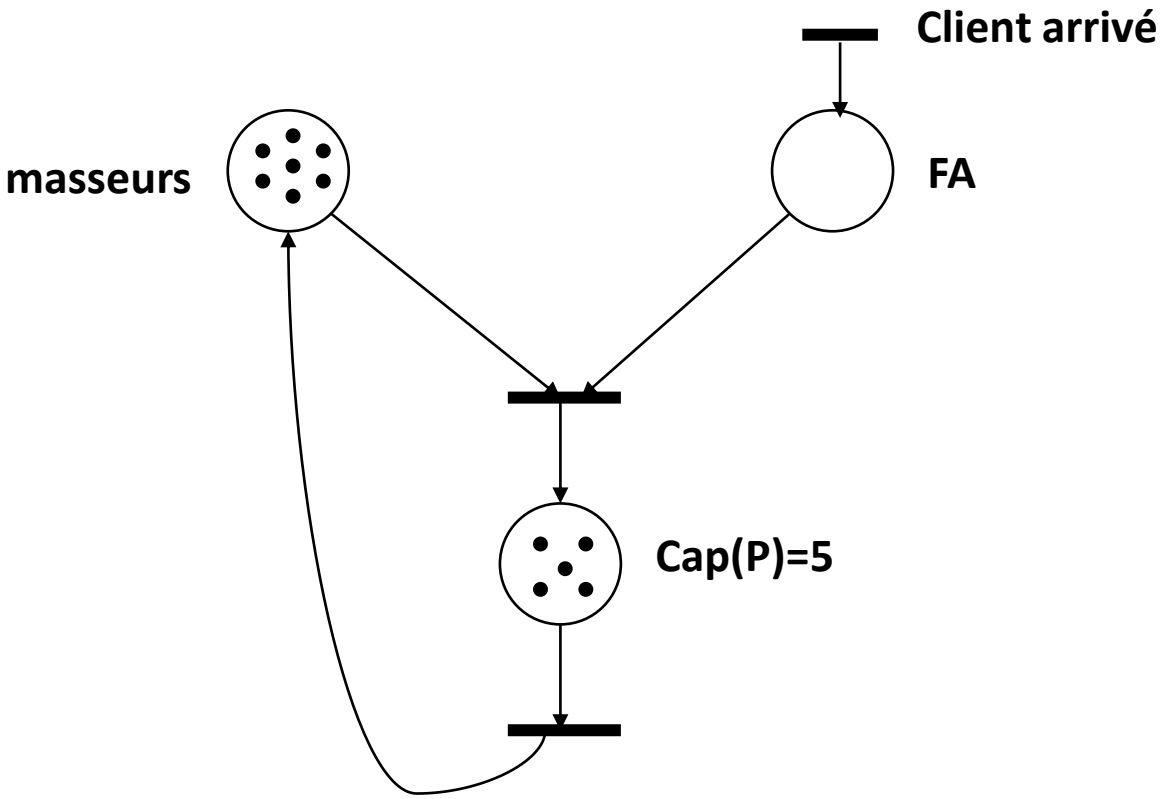
Légende:
FA: file d'attente
Pi : la porte i

Exercice 2:

Une première solution consiste à modéliser la salle de bain en tant qu'ayant la capacité 1. dans ce cas il faut libérer la salle une fois le bain terminé.

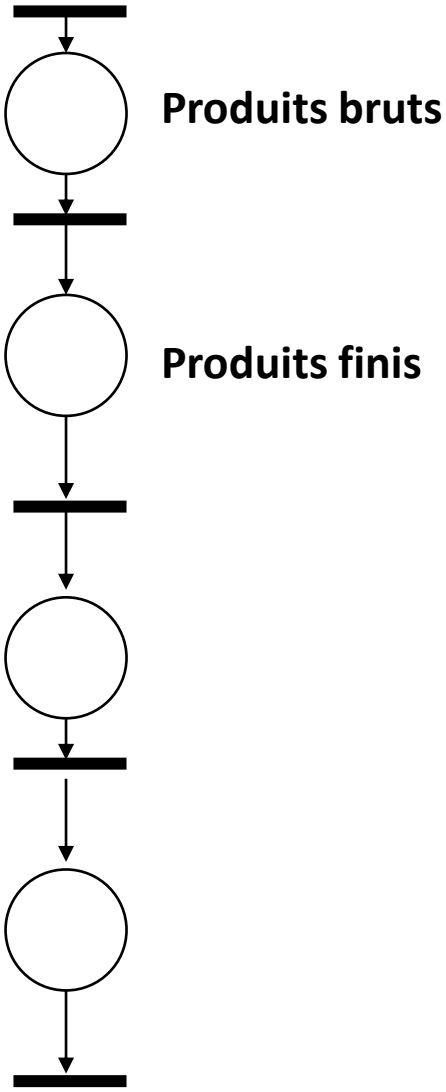


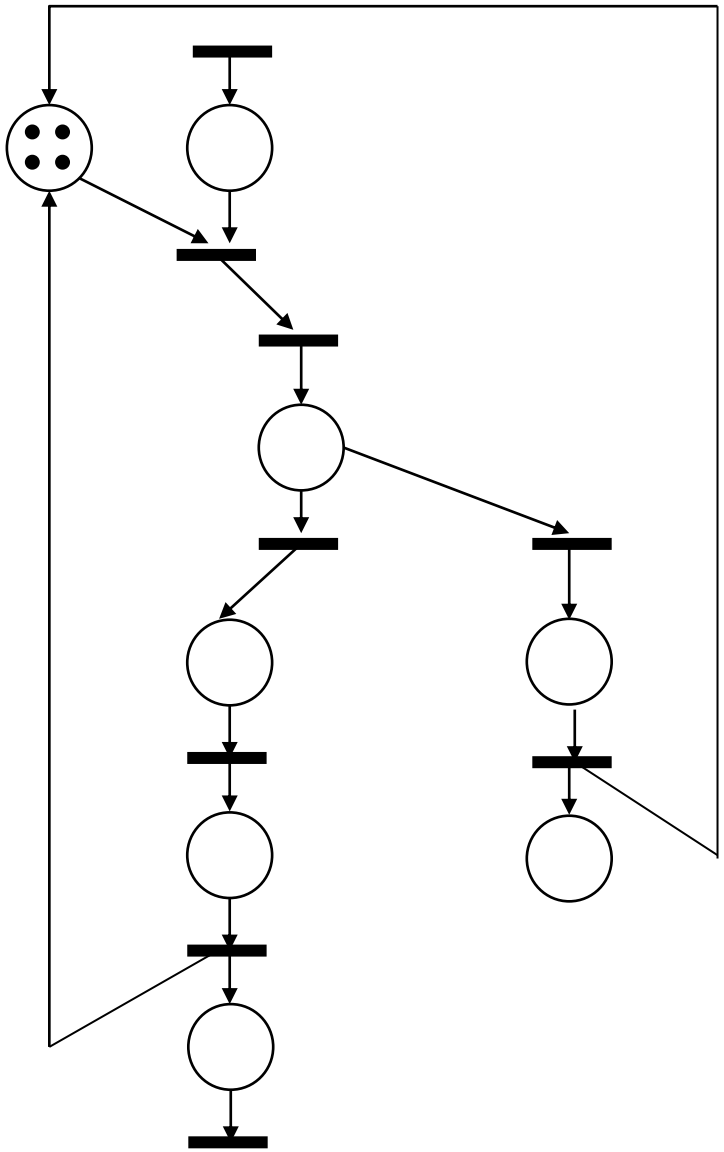
Une deuxième solution consiste à considérer une salle avec la capacité 5 . Dans ce cas ce n'est plus nécessaire de libérer la salle. Cette situation est modélisée par le réseau à capacité suivant:



Exercice 3:

Le système peut être décrit de manière générale par le RdP suivant qui modélise la chaîne de production.





جامعة محمد الخامس السويسي
Université Mohammed V - Souissi

ENSIAS

Corrigé TD4

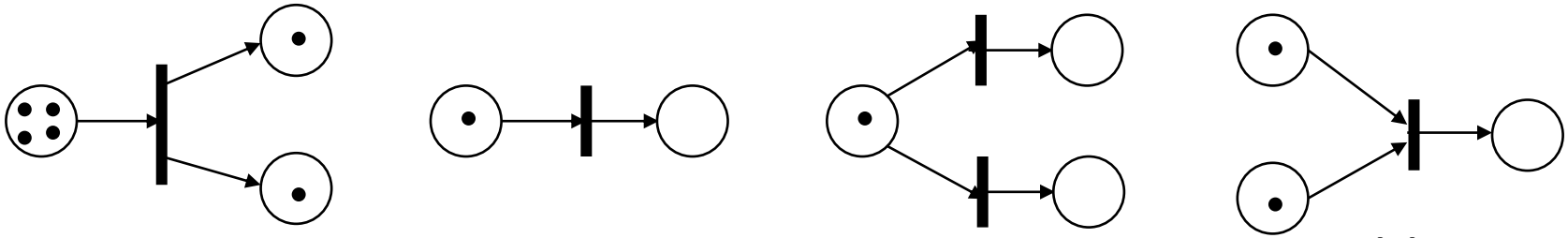
[modélisation à l'aide des réseaux de Pétri]

Systèmes de production & RdP colorés

énoncé

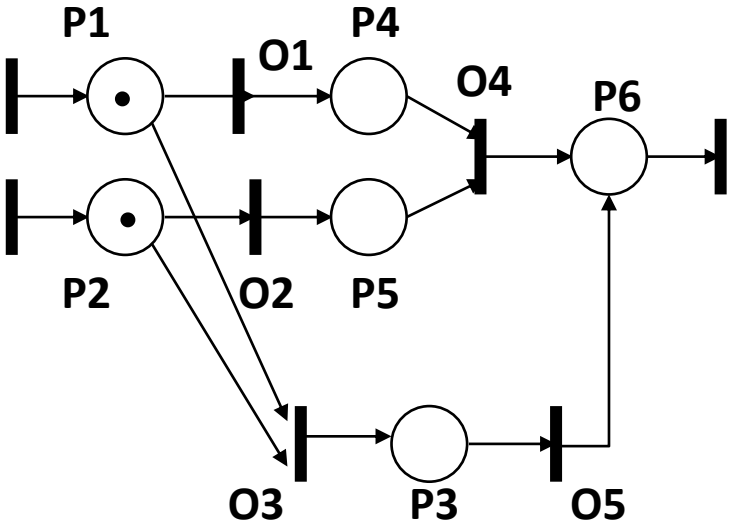
Exercice 1:

Donner l'interprétation des formes suivantes dans les réseaux de pétri décrivant un système de production.



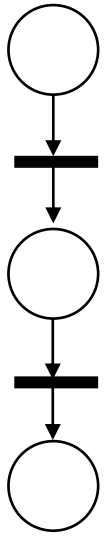
Exercice 2:

Interpréter le RdP suivant (toujours au cadre d'un système de production)



Exercice 3:
modéliser le fonctionnement d'un producteur-consommateur avec un RdP. Initialement, le producteur est en train de produire un objet, le consommanteur en train de consommer et le stock a une capacité de 3 places.

Exercice 4: RdP colorés
Tranformer le RdP ordinaire suivant en un RdP coloré.



Exercice 1:

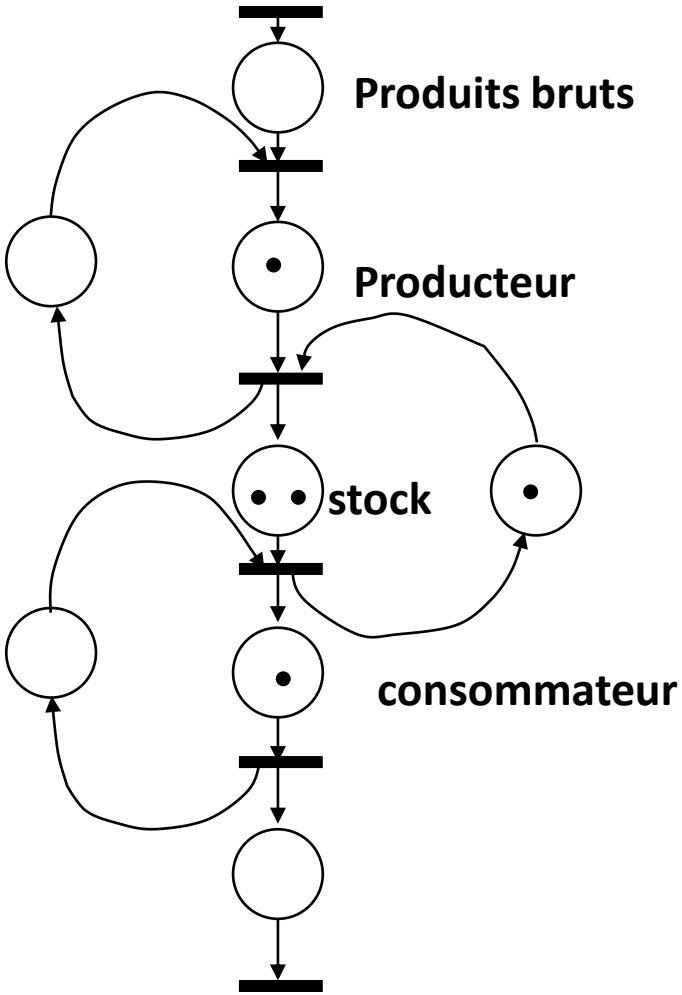
(1) Désassemblage
(2) Transformation
(3) Choix
(4) assemblage

Exercice 2:

On peut obtenir le produit final de deux manières différentes: d'abord l'assemblage (opération O3) puis la transformation (opération O5), ou bien la transformation de chacun des deux composants à part (opérations O1 et O2) puis l'assemblage (opération O4). Toutefois, on est pas toujours sûr d'obtenir le même produit final en suivant les deux chemins .

Exercice 3:

On peut modéliser ce fonctionnement par un RdP à capacité ou l'on consacre une place à capacité 1 pour le producteur, une autre au consommateur et une place pour le stock de capacité 3. le RdP suivant est un RdP ordinaire équivalent au RdP à capacité décrit ci-dessus .


Produits bruts
Producteur
stock
consommateur

جامعة محمد الخامس السويسي
Université Mohammed V - Souissi

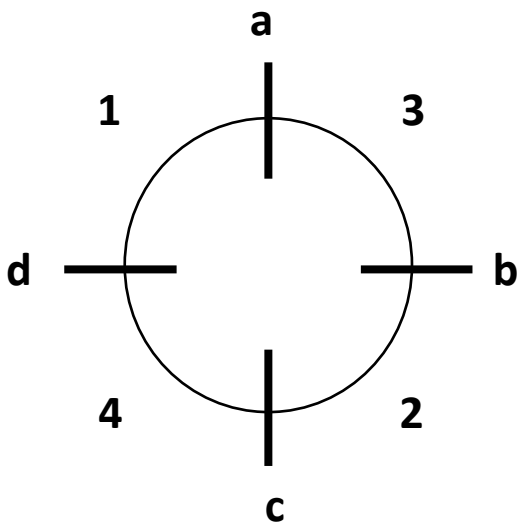
ENSIAS

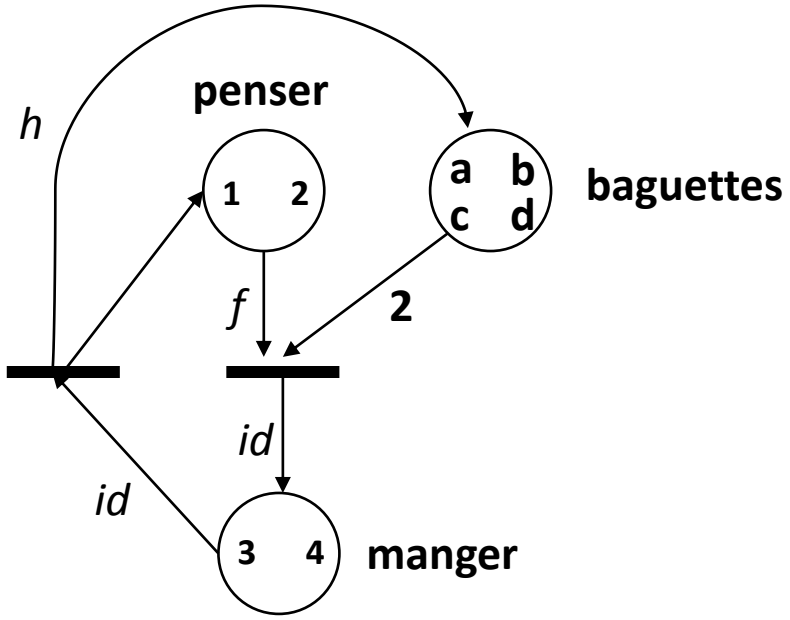
Corrigé TD5

[modélisation à l'aide des réseaux de Pétri]

RdP colorés

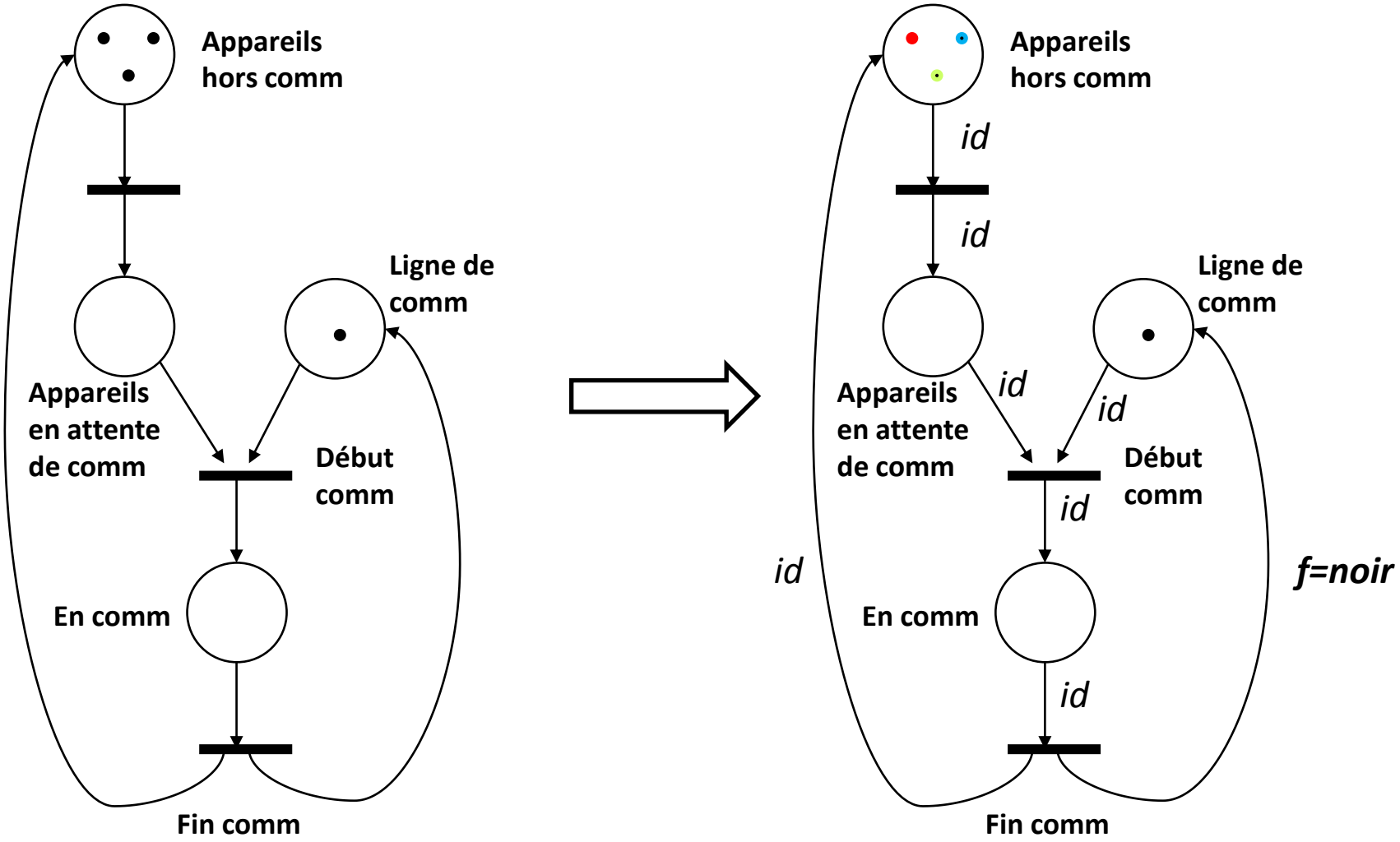
Il faut distiguer entre les philosophes d'une part, et les baguettes d'une autre part. en effet un philosophe donné ne peut pas faire usage de n'importe quelle baguette, aussi deux philosophes adjacents ne peuvent pas manger en même temps.





f(1)= a+d
f(2)= c+b
f(3)= c+d
f(1)= a+b

Exercice 2:


Appareils
hors comm
Ligne de
comm
Appareils
en attente
de comm
Début
comm
En comm
Fin comm
Appareils
hors comm
id
id
Ligne de
comm
Appareils
en attente
de comm
id
id
Début
comm
id
id
En comm
id
id
f=noir
Fin comm

Exercice 3: