Théorie et pratique de la concurrence

CM6 TD5 Producteur - Consommateur

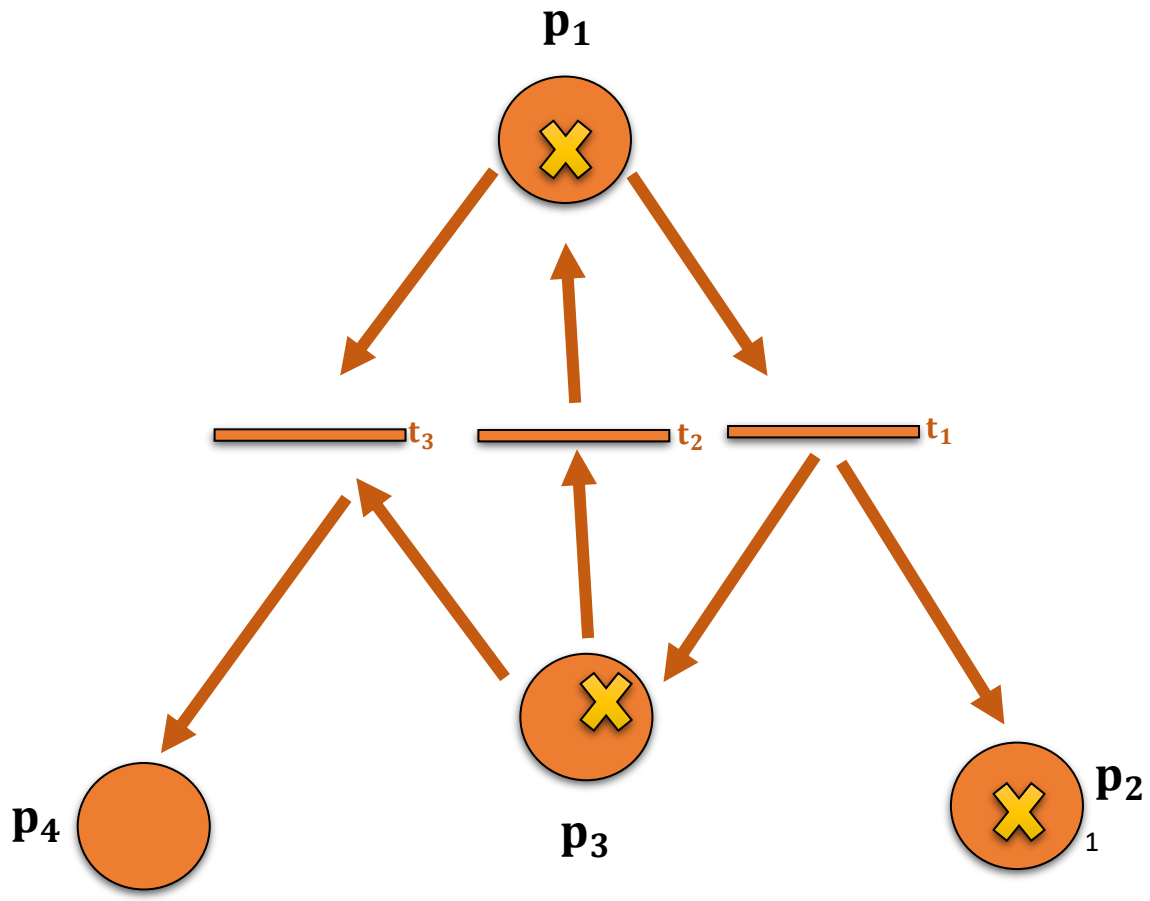
Lundi 18·10·2021

Exercice • Marquage accessible, mauvaise situation

- $\exists\, M$ accessible $t \cdot q \cdot\ M(p_4) \geq 2$?

  Le fait d'avoir 2 jetons à la place P_4 est considérer comme une « mauvaise situation ». Cette situation peut arriver ou pas ?

- $(3, 1, 1, 0)$ est accessible ?



$(1, 0, 1, 0)$

t_1 ; t_2 ; t_3

$\leq$

$(0, 1, 2, 0)$ $\quad$ $(2, 0, 0, 0)$ $\quad$ $(0, 0, 0, 1)$

blockage

t_2 $\quad$ t_1

$(1, 1, 1, 0)$

Deja visité

$(1, \underbrace{1}_{w}, 1, 0)$

non – borné

On voit que le marquage initial est plus petit ou égal (≤) que ce marquage. Donc, la place p_2 qui est strictement plus grand devient ω (oméga). Ce n'est pas borné mais faut continuer à développer sauf si on reutembe sur un sommet déjà vu et alors on arrete.

t_1 $\quad$ t_2 $\quad$ t_3

$(0, \omega, 2, 0)$ $(2, \omega, 0, 0)$ $(0, \omega, 0, 1)$

blockage

t_2

$(1, \omega, 1, 0)$

Marqué

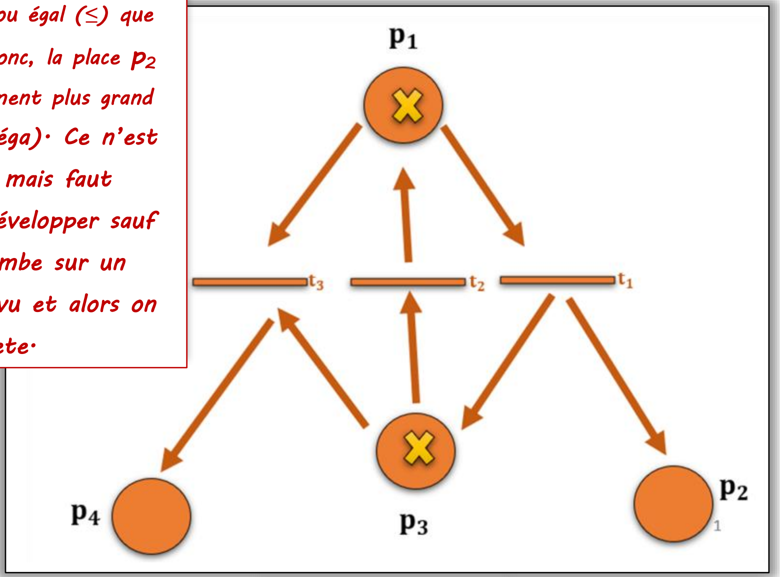
On ne va pas faire d'extra-polation car on tombe sur lui même $(2, \omega, 0, 0) \geq (2, 0, 0, 0)$

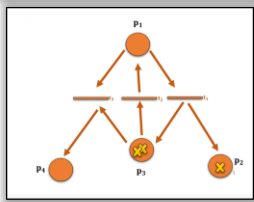
t_1

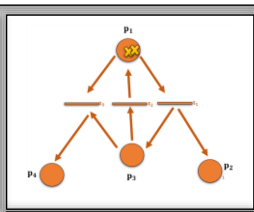
$(1, \omega, 1, 0)$

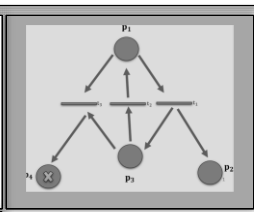
Marqué

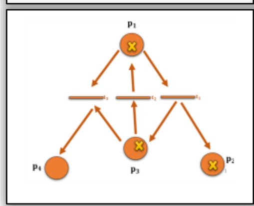
Il y aura des accessible avec une valeur (1, ?, 1, 0)

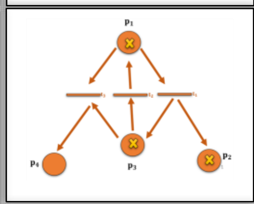


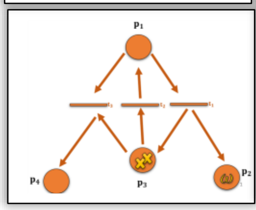


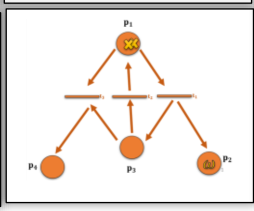


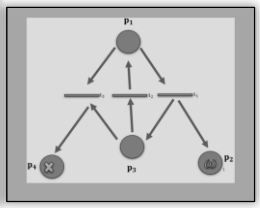


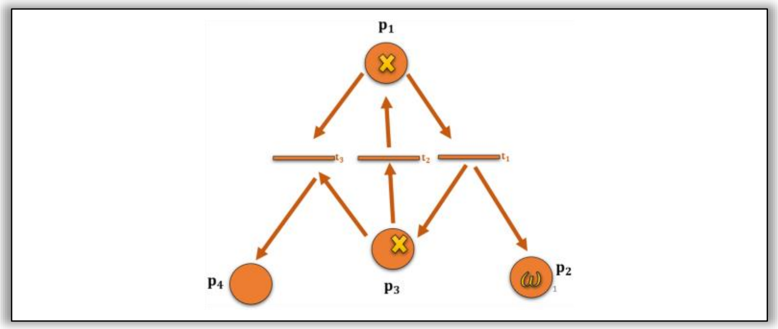


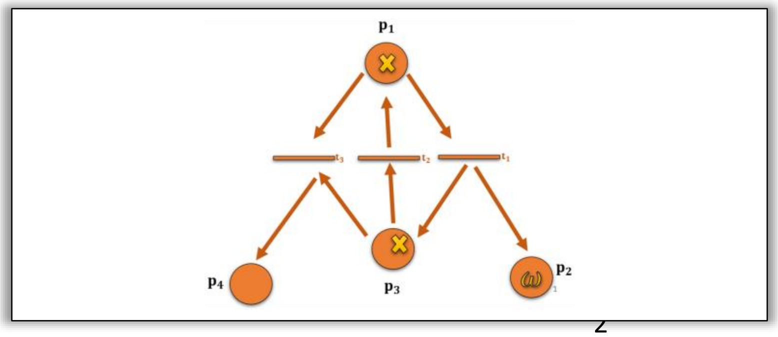










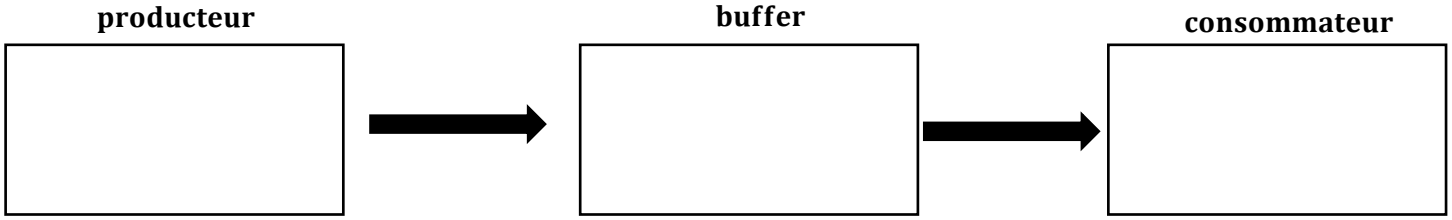


- Les valeurs de p_4 c'est soit 1, soit 0. Donc, c'est claire que $M(p_4) \geq 2$ c'est PAS possible.
- **Pour p1 on a 0 / 1 / 2 mais jamais 3, donc la reponse est que (3, 1, 1, 0) est pas acessible.**

Producteur - Consommateur

On a un producteur et un consommateur.

Producteur - il travail, il fabrique des choses, et ensuite il va envoyer ca vers le consommateur à travers un buffer.



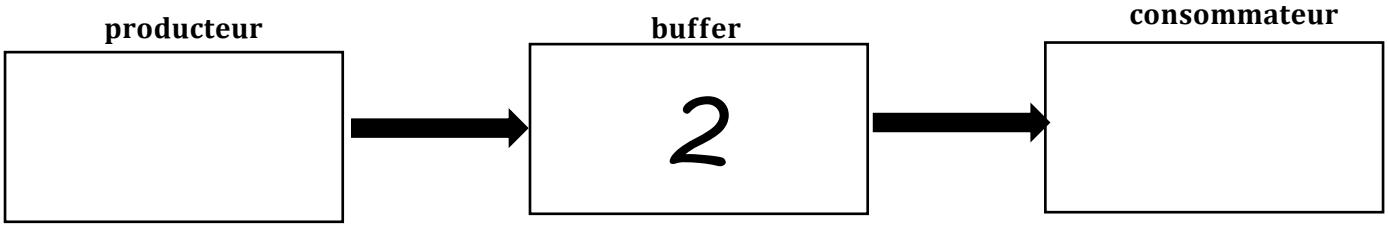
Plus le buffer est petit, plus le producteur devra aller longtement.

Types d'ordonnonceurs pour les buffers

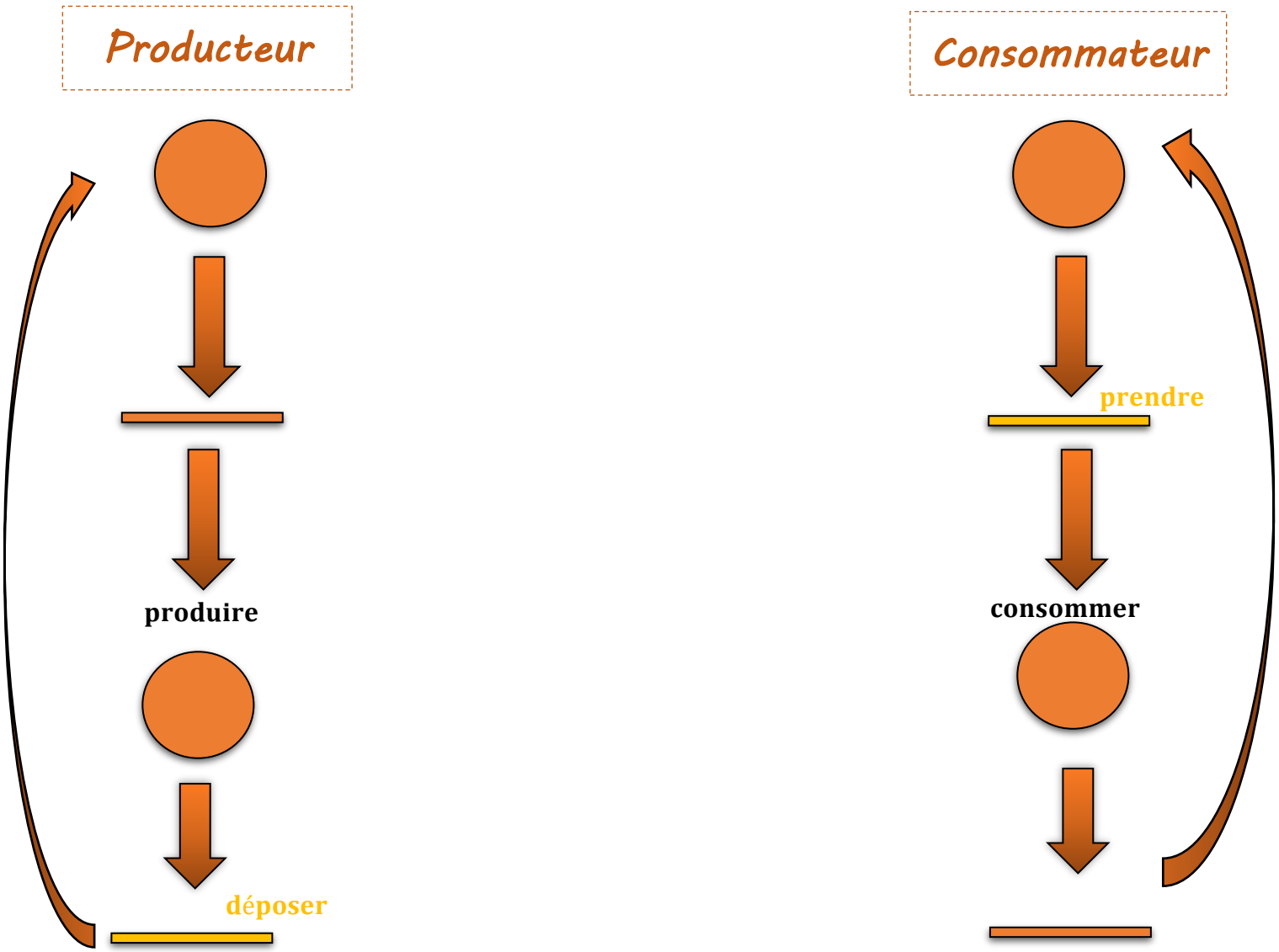
- Non-ordoné
- FIFO
- LIFO

Exercice

Faire un reseau de Petri pour chaque un des 3 cas de buffers (non-ordonné, FIFO, LIFO), taille du buffer : 2 place.



- Conseil : suivre l'architechture du système.



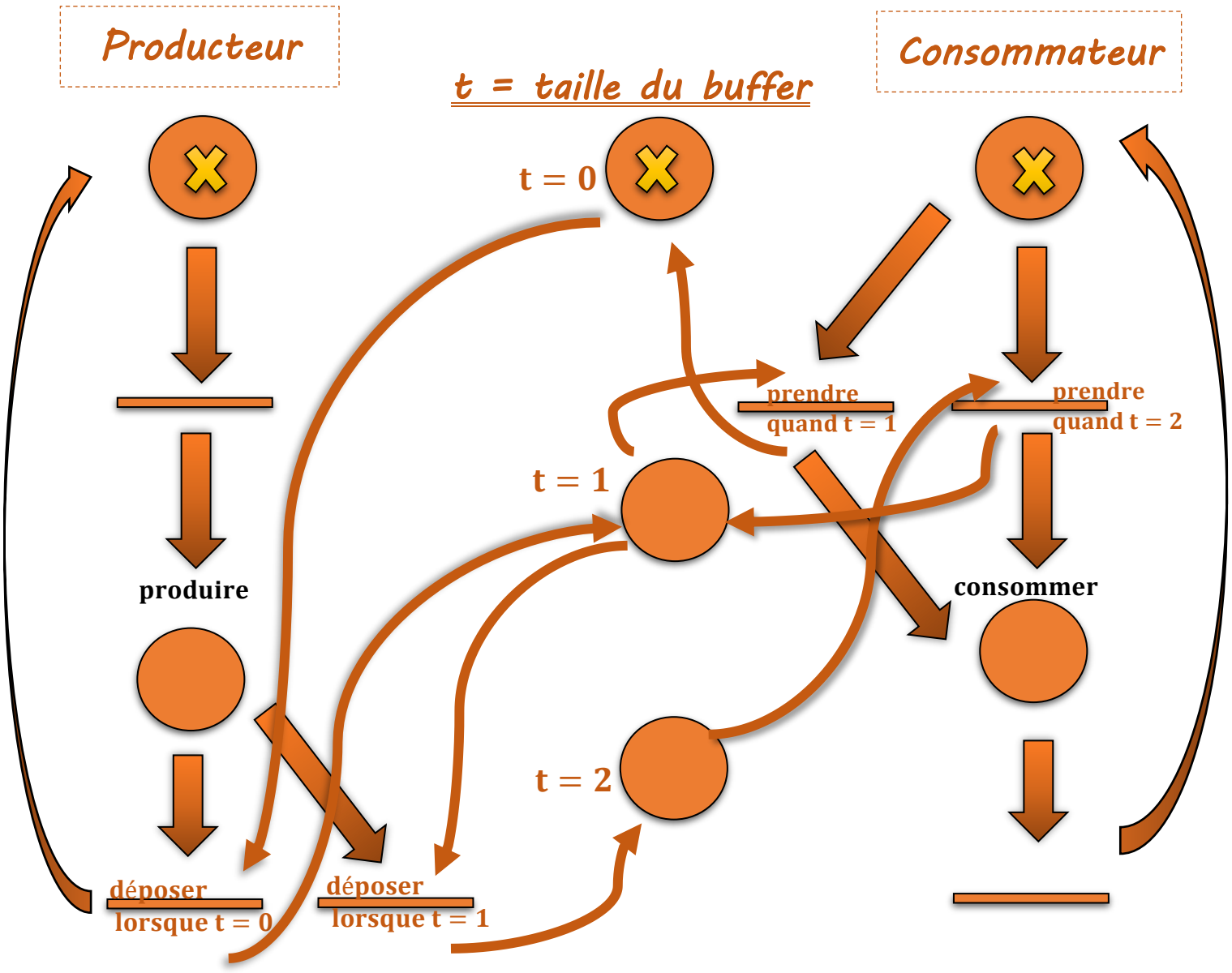
En jaune : les transitions critiques.

Quelle sont les conditions nessaireres pour realiser ces actions ?

- **Pour qu'on puisse déposer :** le buffer doit avoir une taille de 0 ou 1. Suite a la transition, le contenu du buffer va acrémenter (à 1 ou à 2).
- **Pour prendre :** le buffer doit avoir une taille de 1 ou 2.
  Suite a cette transition, le contenu du buffer va décrémenter (de 2 à 1 ou de 1 à 0).

Cas Non-ordoné

- **Dans ce cas, l'ordre du contenu du buffer ne m'intéresse pas.**
- *Possible qu'il y aura des modifications légéres a faire aux chémas precedents.*



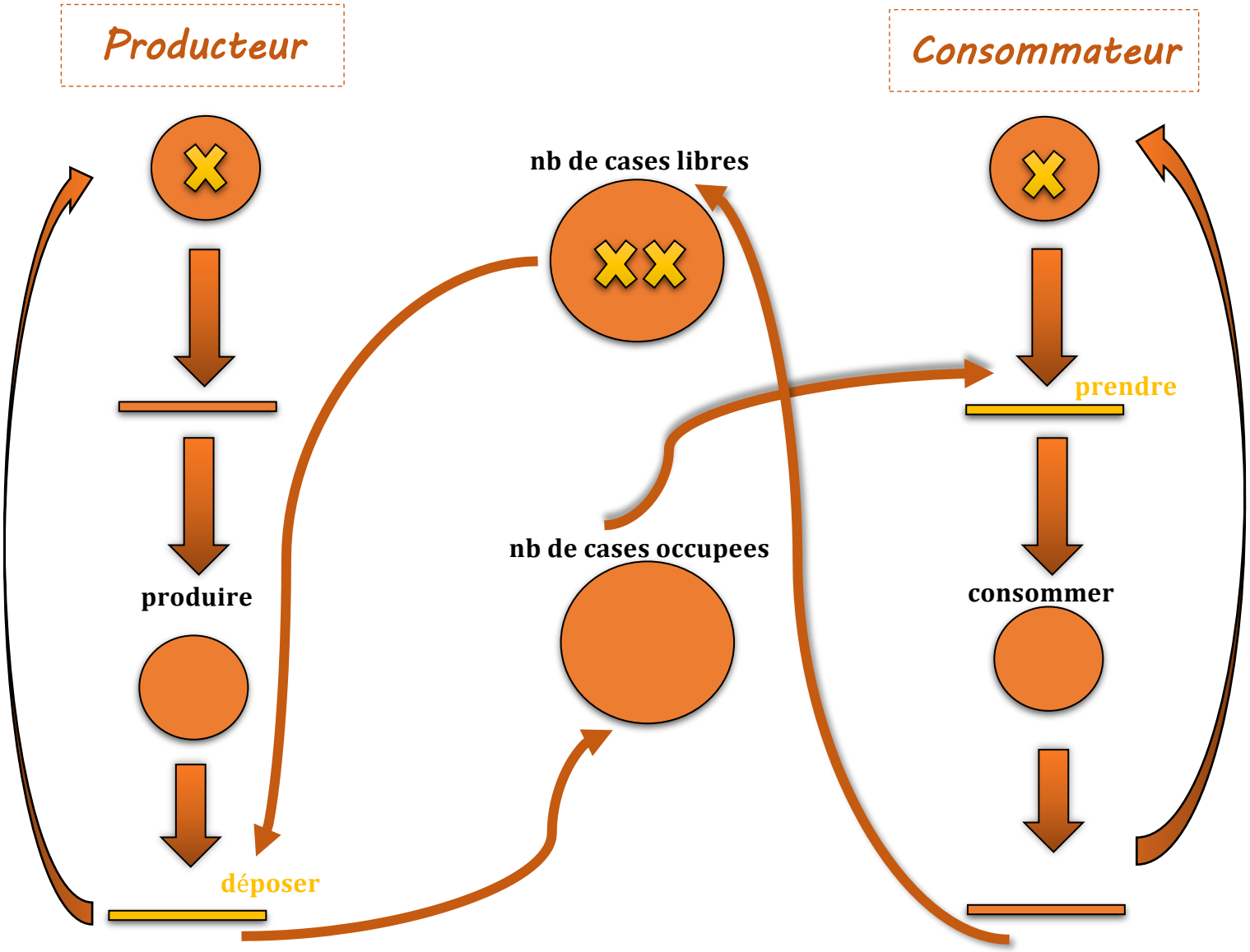
On a besoin de savoir 3 informations :

- *Le beuffeur est de taille 0 ? (pas rempli)*
- *Le beuffeur est de taille 1 ?*
- *Le beuffeur est de taille 2 ?*

- **La taille est borné à 2.**

Autre solution

Compter le nombre de cases libres et le nombre de cases occupés.

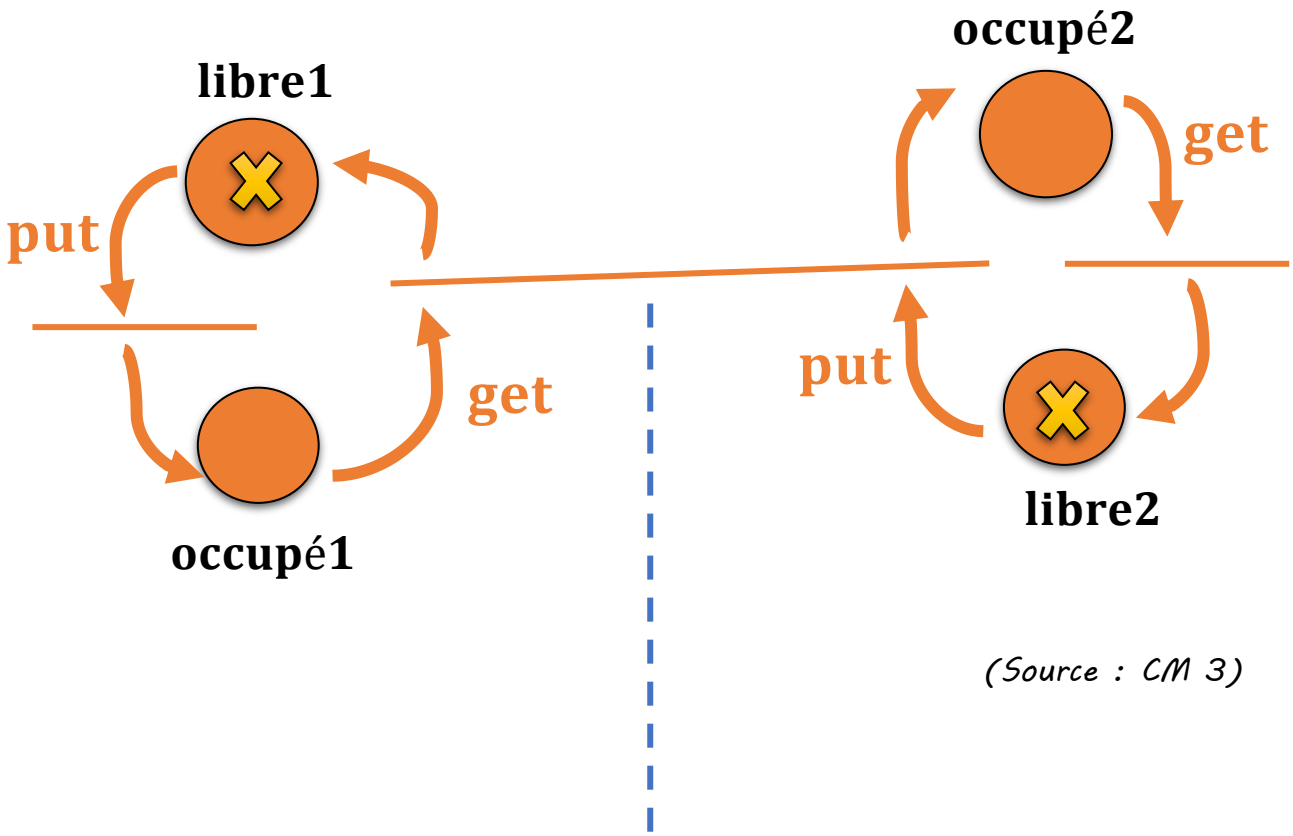


En jaune : les transitions critiques.

En effet, c'est la meme idee que la solution precedente, sauf que a la solution precedente on ses limité a 1 jeton max par case. La on peut avoir plus que 1 jeton par case, ce qui nous permet de simplifier, mais l'idee est la meme.

Cas FIFO

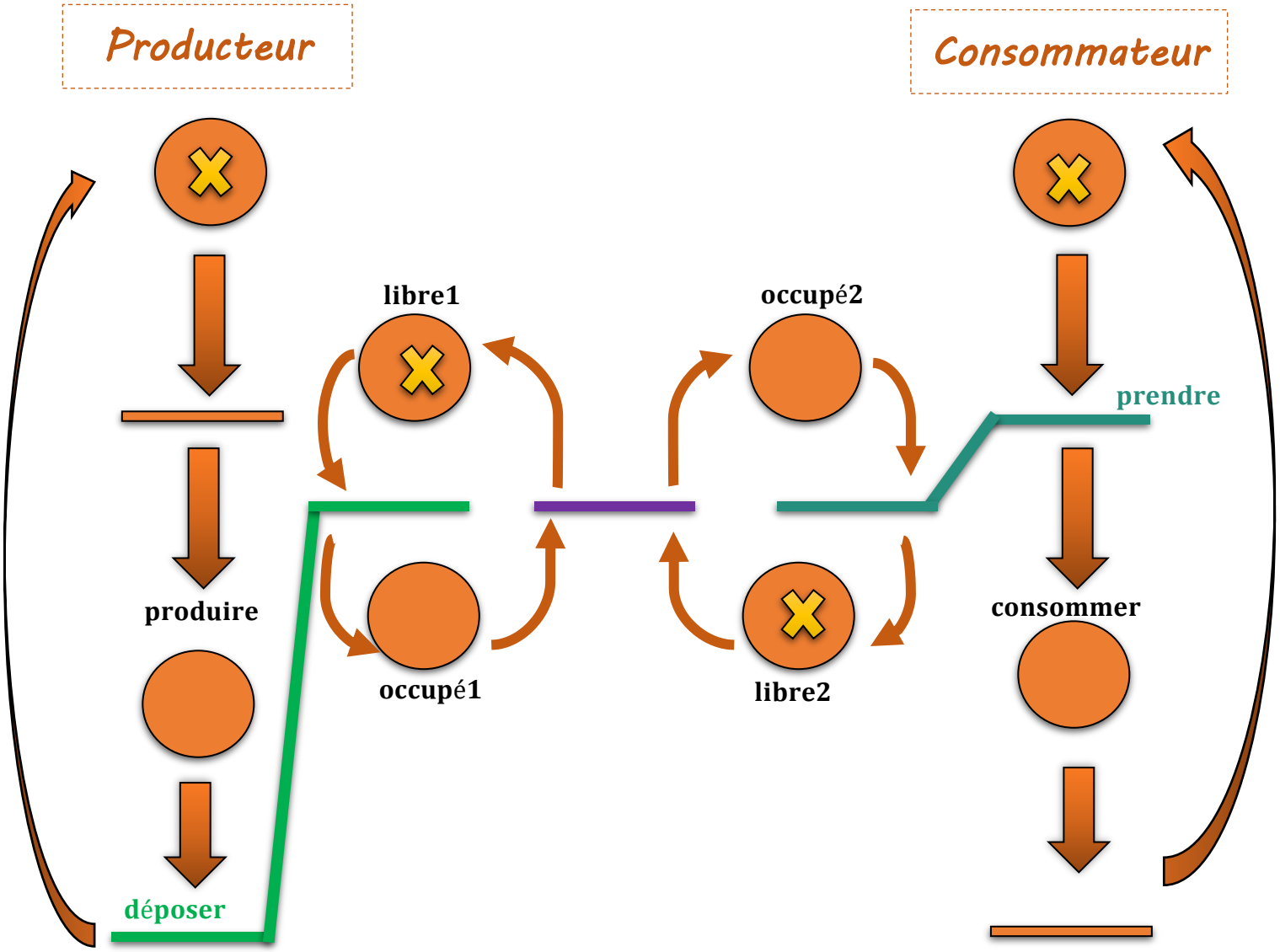
Au debut de l'anée on a déjà vu le cas d'un buffer FIFO à 2 places :



(Source : CM 3)

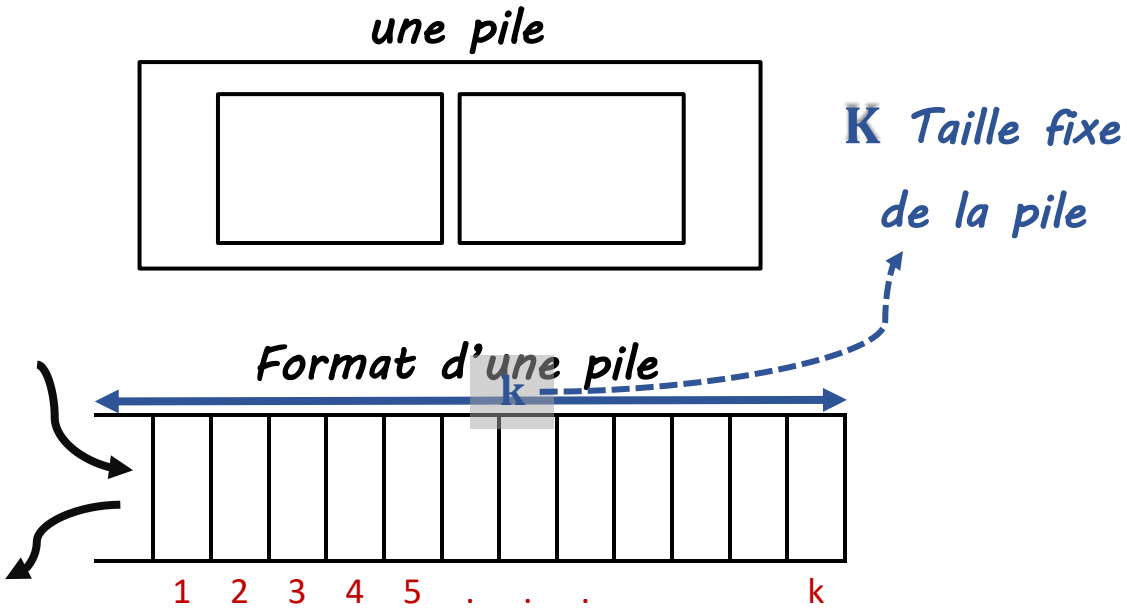
La transition ou ils vont se synchroniser

A cela, on ajoute notre producteur et notre consommateur.


Producteur
Consommateur
libre1
occupé2
prendre
produire
occupé1
libre2
consommer
déposer

Cas LIFO

- Faut PAS changer la partie consommateur et la partie producteur.
- Refleshir comment un LIFO tous seul marche, et comment metre les 2 en parallelles (2 buffer LIFO en parallelle).



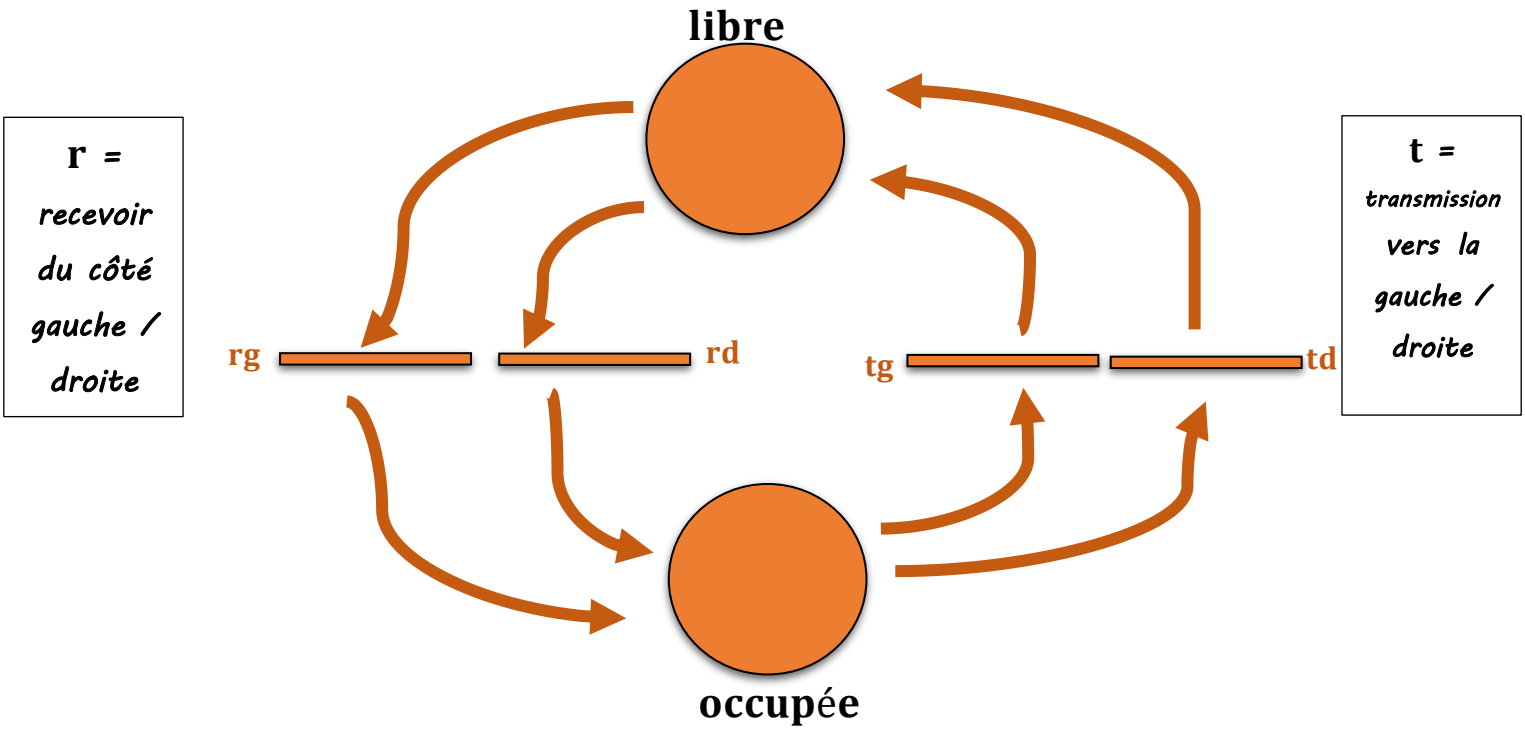
- On veut communiquer qu'avec le haut de la pile.

Le comportement de chaque cellule

- Si elle est libre, elle peut recevoir de l'information fraiche du côté gauche ou recevoir de l'information ancienne du côté droit.
- Si on me demande « donner moi quelque chose », c'est que le dernier élément inséré qui va sortir.

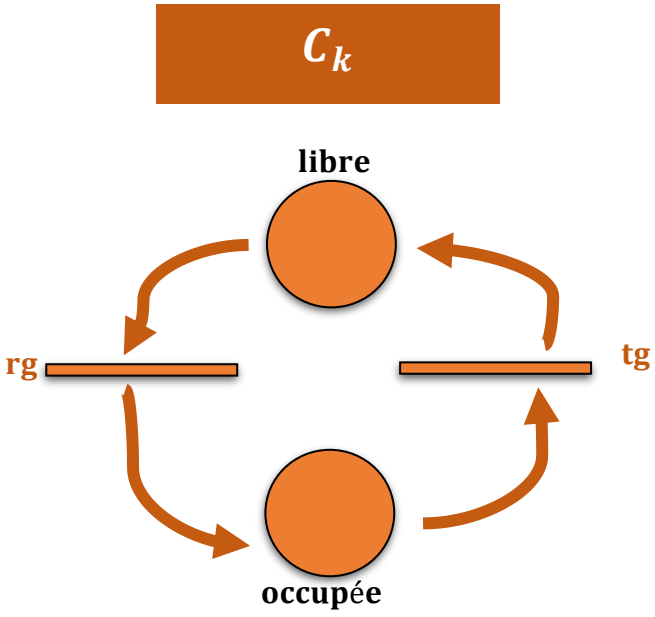
Les cellules sont ordonnées de 1 à k. Prenons une cellule quelle qu'onques de l'intérieur de la pile : la cellule i

$i-1$ C_i $i+1$



Est-ce que la cellule du début et la cellule de la fin on le même comportement que la cellule « i » ?

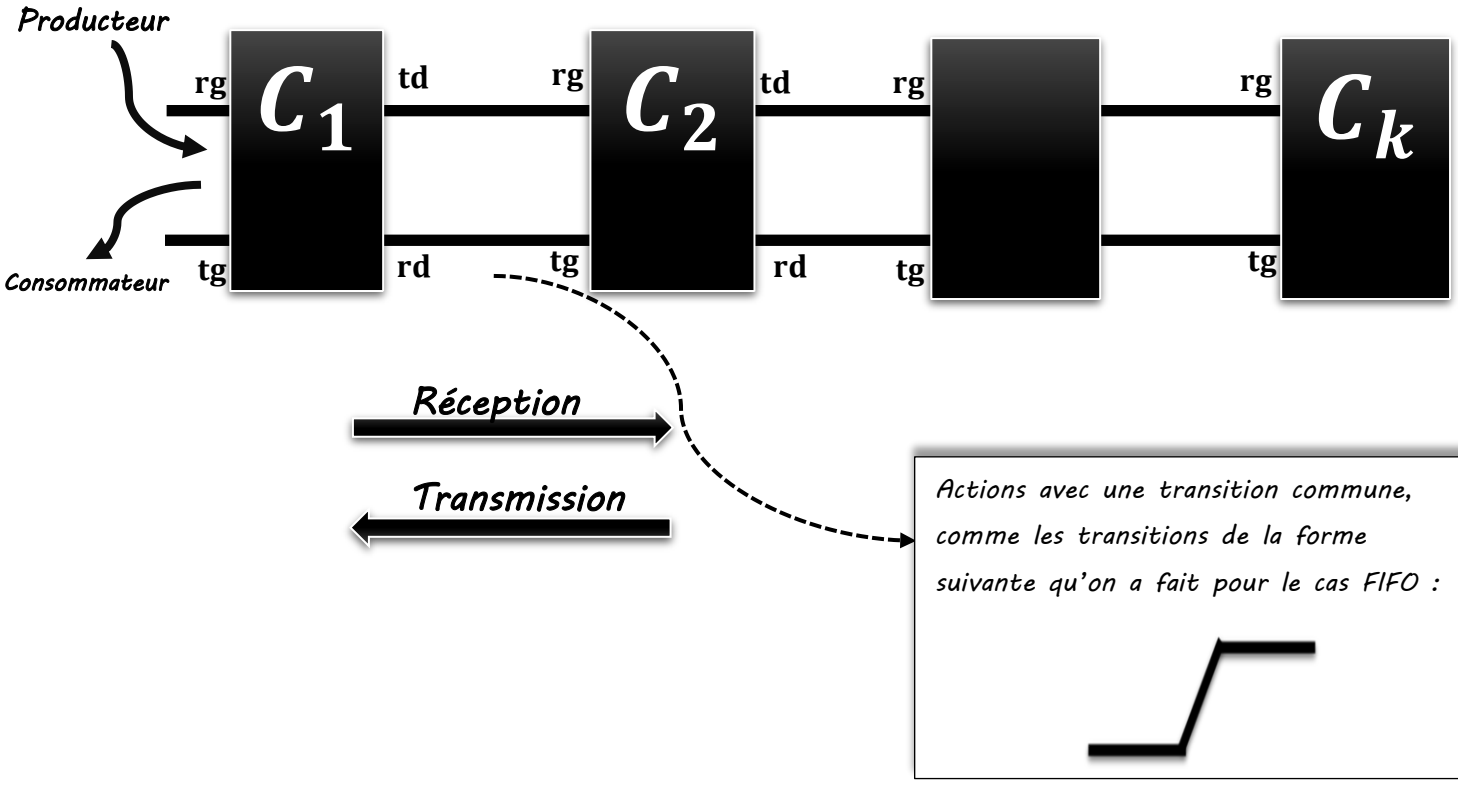
La cellule au fond, la dernière, peut que recevoir ou rendre. On va l'appeler la cellule C_k



Pour la première cellule (le haut de la pile) c'est différent que pour la cellule « i » ?

Non, c'est pareil ! Y'a que la cellule k qui est différente.

Mtn, comment connecter les cellules ?



Exercice • La piscine

On va considérer un system pour gérer une piscine et il y a des personnes qui arrives :

1. **Première chose :** chercher la clé d'une cabine pour se changer. Le nombre de cabines est limiter.
2. **Une fois la clé obtenue :** la personne doit aller se changer et poser ses vêtements dans un panier.
3. **Rendre la clé de la cabine**
4. **Accéder au bassin (la piscine)**
5. **Récupérer le panier et chercher la clé d'une cabine.**
6. **Rendre la cabine et le panier.**

La première étape : chercher une cabine (= chercher la clé, c'est pareil).

- Les clés : une ressource qu'on modélise avec une place qui contient k-jetons.

Et si je veux un comportement plus général ou l'ordre (clé / panier) est pas imposé. Comment le modéliser avec les réseaux de Petri ? (Les arcs en rouge).

Le comportement initial, qui est mauvais, reste toujours. Mais mtn, il y a un ajout de nouveaux problèmes ?

Oui, un problème de blocage, mais différent.
1er arriver : prend la clé -> le 2eme commence par l'autre ordre et prend le panier. Résultat : blocage.

Donc , chaque fois qu'on programme de manière parallèle : faire attention au blocages.

Exemple avec 1 personne