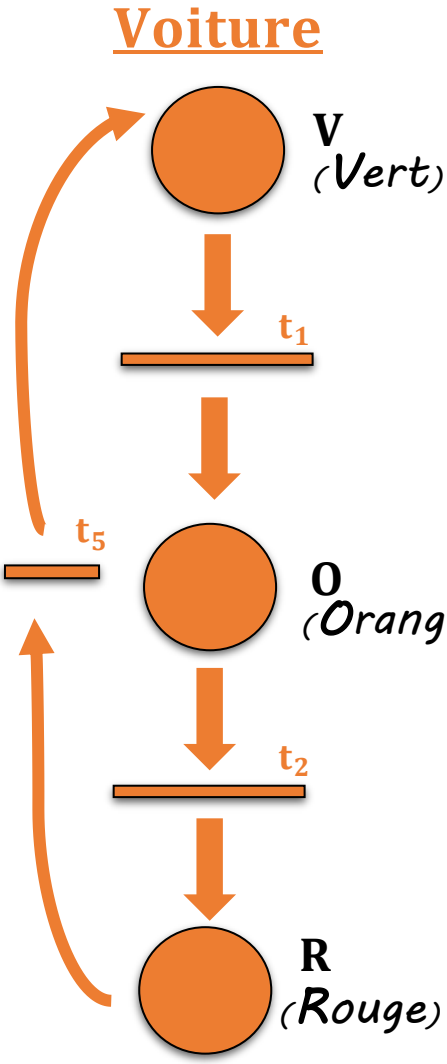
TD2 Réseau de Petri

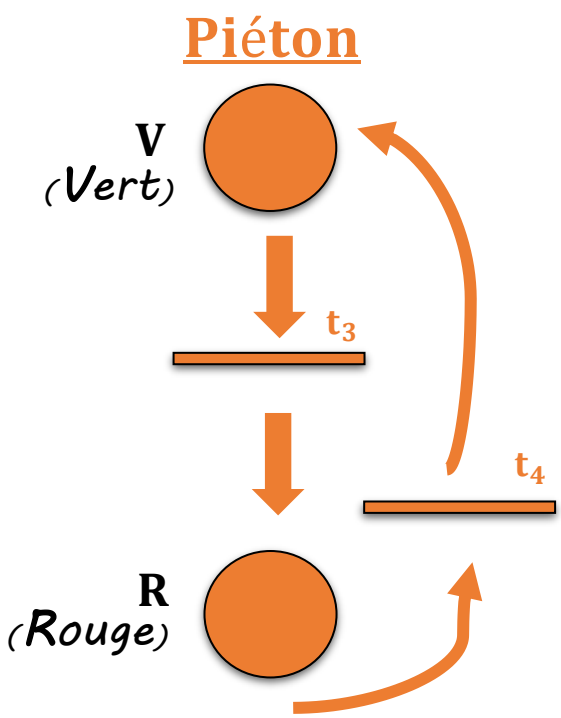
Lundi 27·09·2021

Exercice 1 • Feu rouge / vert

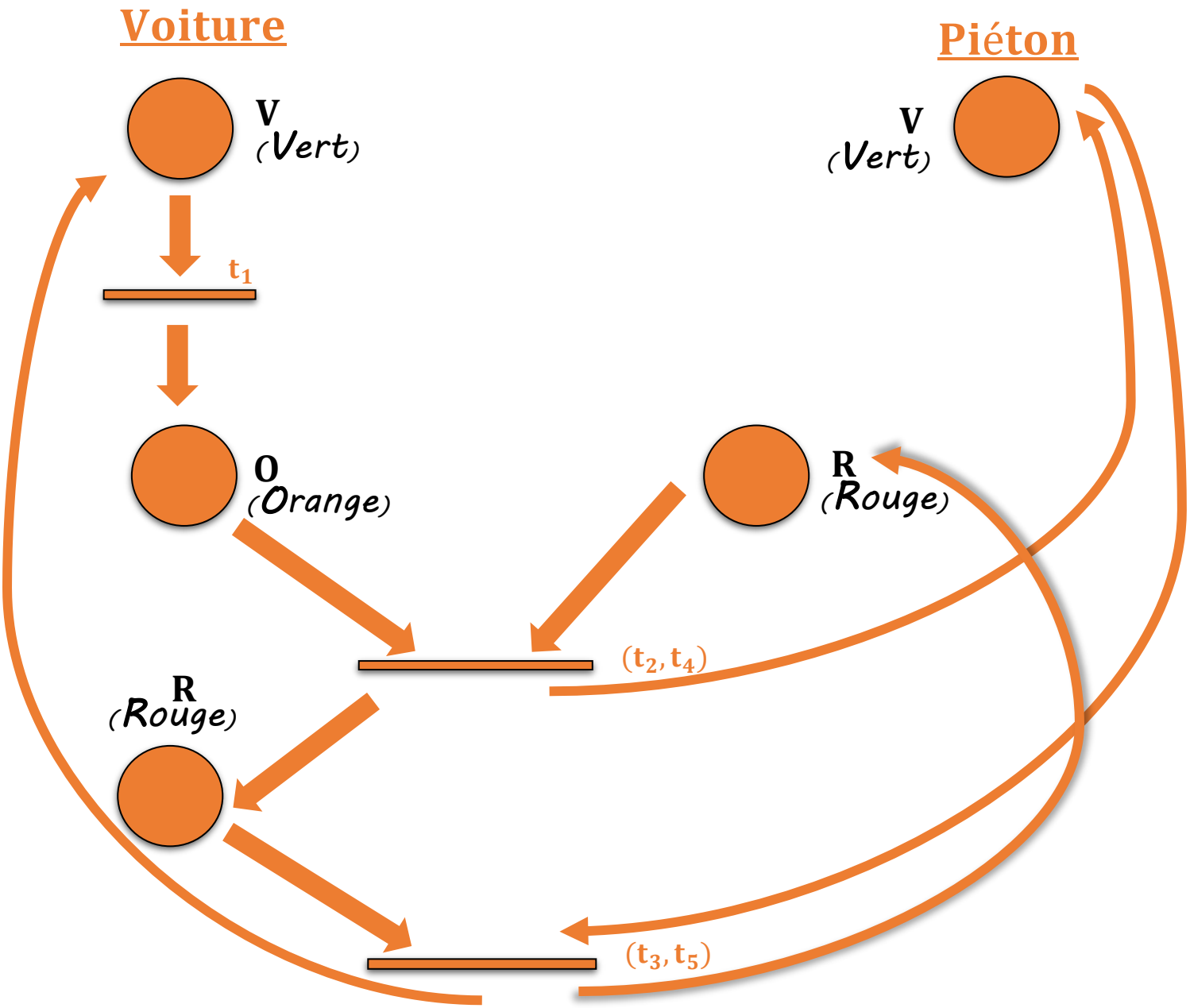
- Feu pour voiture : rouge, orange, vert·
- Feu pour piéton : rouge, vert

Modéliser les feux·





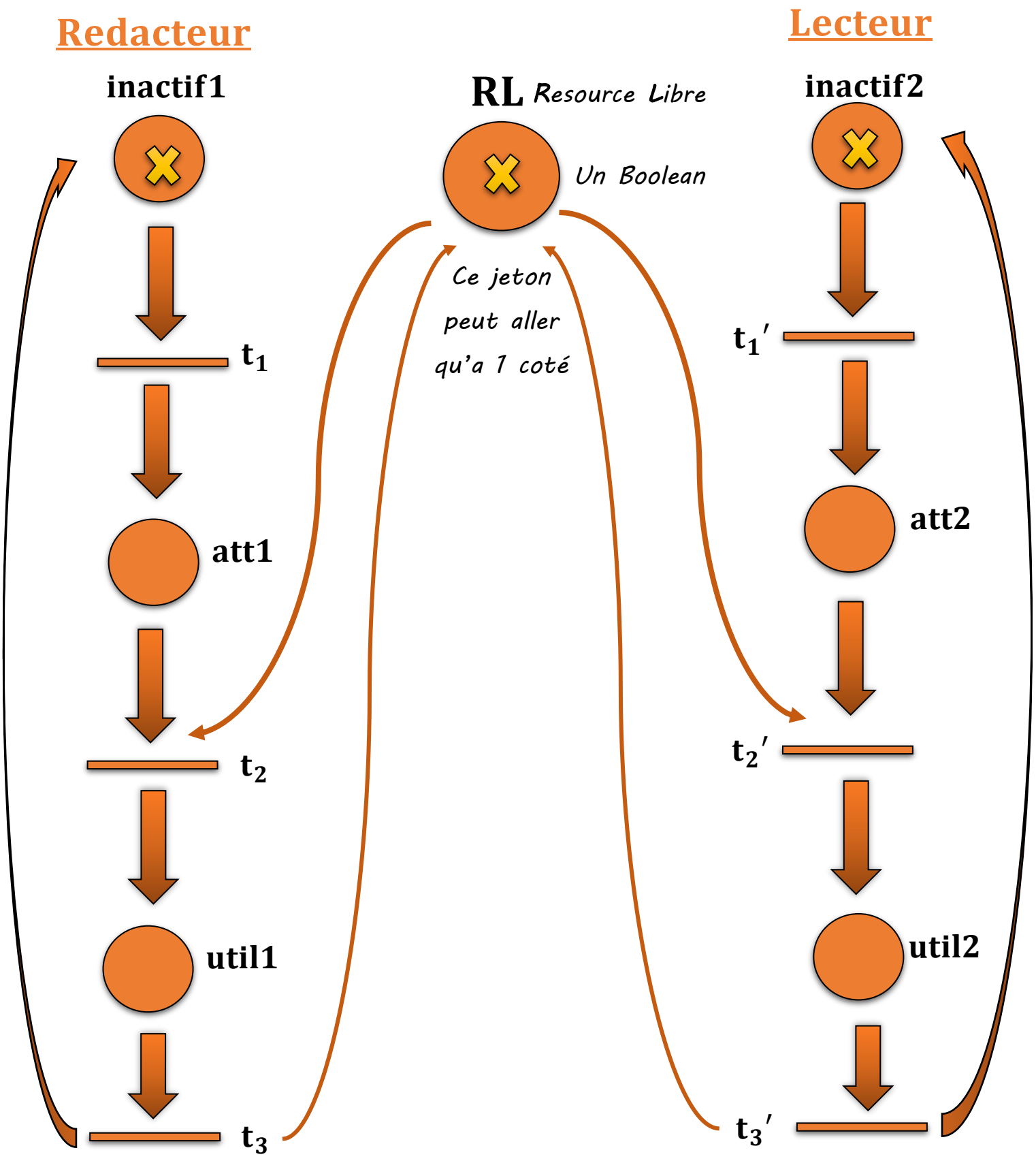
Maintenant, on va essayer de combiner, de merger, certaines transitions.



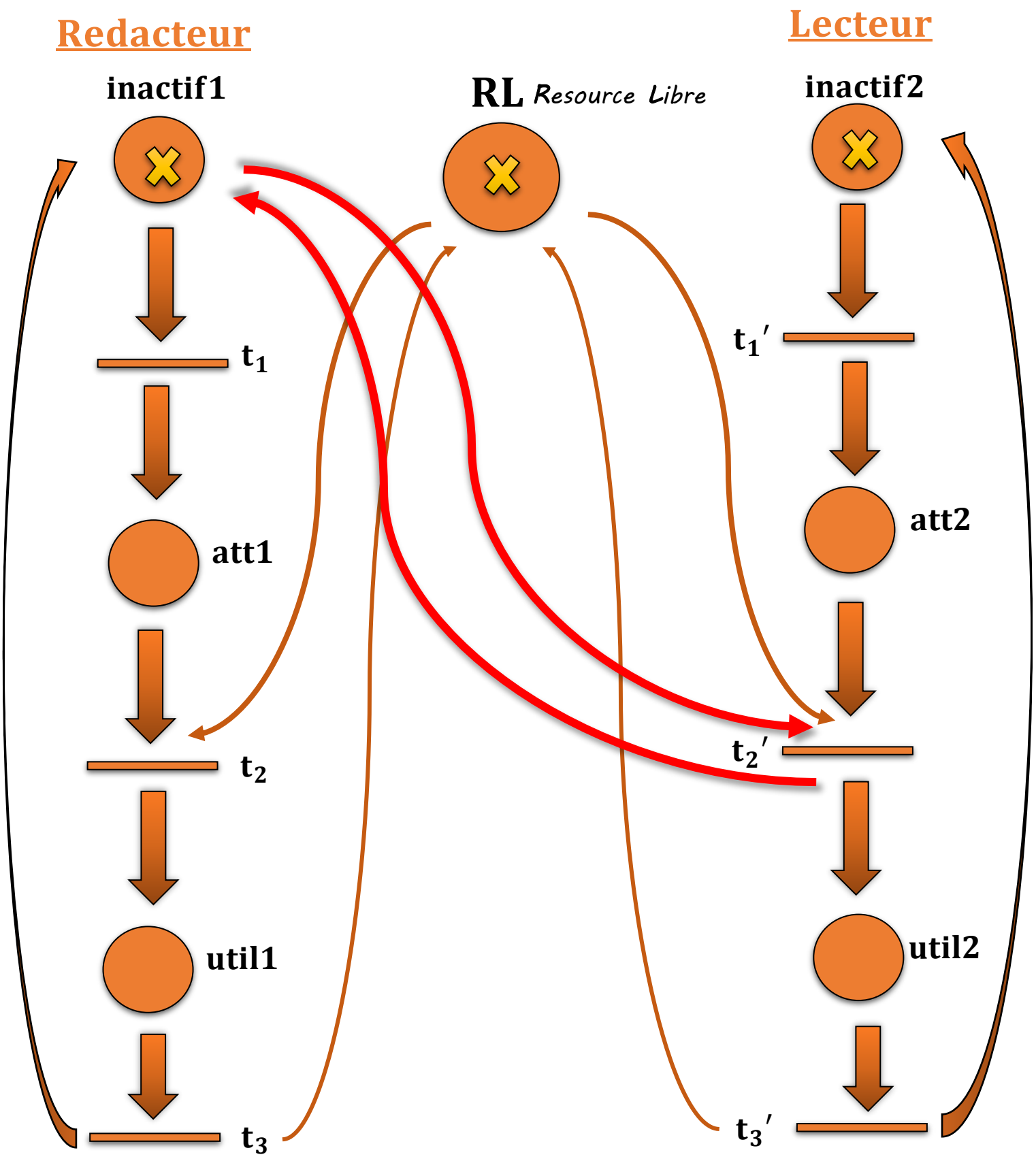
Exercice 2 · Lecteur - rédacteur

- On revient au problème du lecteur - rédacteur vu en cm (CM 3)·
- On suppose qu'il y a 3 états pour chaque processus :
  - Pas intéresser par la ressource (il fait autre chose)·
  - Intéresser (attente)
  - Utiliser (et quand il termine, il redevient « intéresser »)·
- Supposons qu'on veut donner la priorité au rédacteur· C·-à-d· :

  Quand j'ai un lecteur en compétition avec un rédacteur, si on doit choisir entre les deux, c'est le rédacteur qui va l'emporter·
- Si le lecteur se présente et le rédacteur n'est pas intéresser, alors il passe·
- 1 rédacteur, 1 lecteur·



Ça c'est le problème du lecteur – rédacteur tel que vu en CM. **Maintenant, on veut donner la priorité au rédacteur.**



A la place des flèches rouge on peut écrire une priorité :

$$t_2 > t_2'$$

Réseaux de Petri avec Priorité

RdP + relations d'ordre entre les transitions.

Partielle (pas forcément total)

Donc c'est possible qu'il y a des transitions pas comparable.

C.-à-d. :

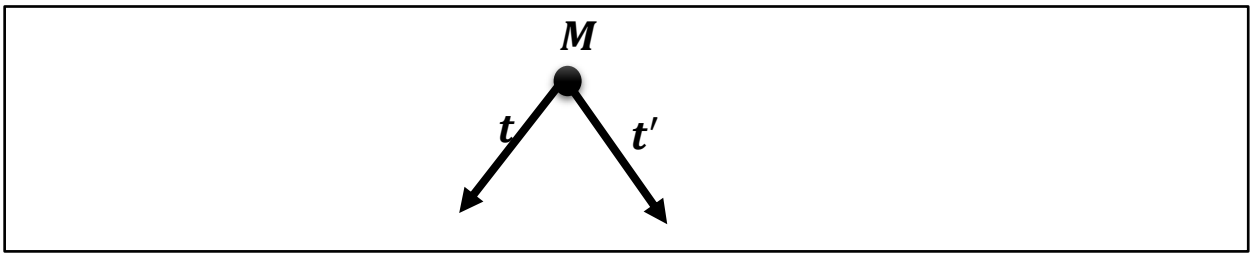
Pour 2 marquages $\mathbf{M}, \mathbf{M}'$ et $\mathbf{t} \in \mathbf{T}$

$$\mathbf{M} \underset{prio}{\triangleright^{t}} \mathbf{M}'$$

Ssi

1) $\mathbf{M} \triangleright \mathbf{M}'$ **Je peux faire cette transition**

2) $\nexists\, \mathbf{t}'\ \ \mathbf{t}' > \mathbf{t}$ et $\nexists\, \mathbf{M}''$ $\mathbf{t.q.}$ $\mathbf{M} \overset{t}{\triangleright} \mathbf{M}''$



Respecter les relations d'ordre lorsqu'elles sont exécutables (si pas exécutable alors non...)

Exercice 3 · Question théorique

De manière général, supposons que notre réseau est un réseau 1-borné (= max 1 jeton par place)· Est-ce que c'est possible que si on nous donne un réseau avec priorité, de le transformer en réseau normal ?

Est-ce qu'on peut démontrer ça ?

(Transformer n'importe quel réseau avec priorité a un réseau normal)

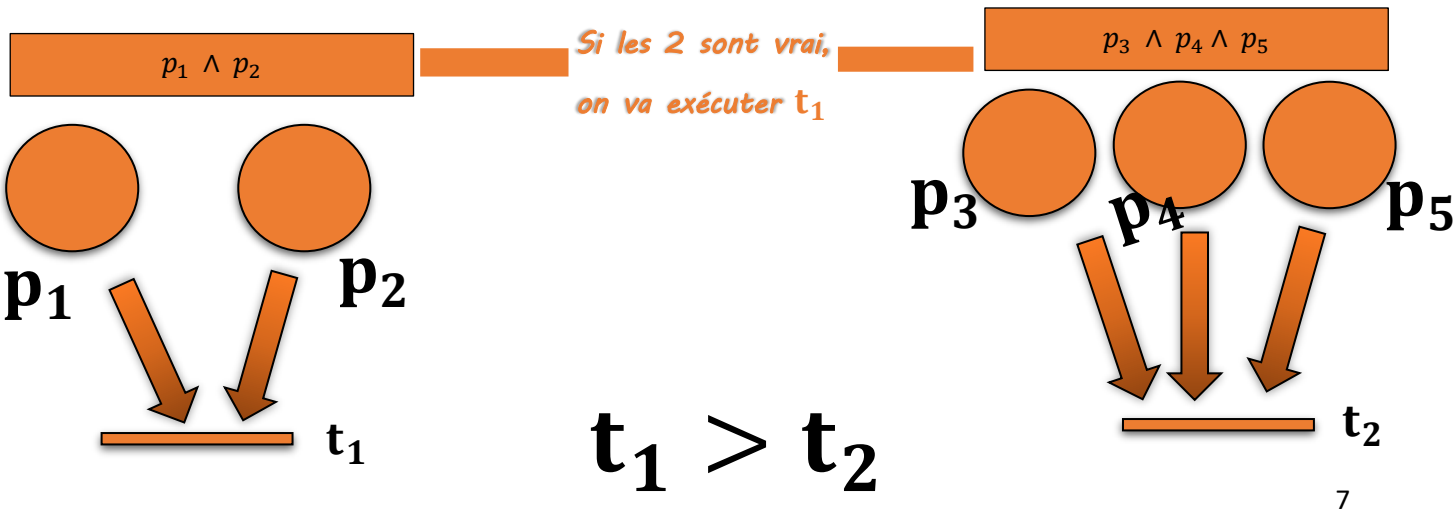
$$t_1 > t_2$$

A chaque fois qu'on doit choisir une des deux, ont choisie t_1

1 – borné

Les places ont comme valeur que 0 ou 1,
c'est donc un Boolean·

Quesque ça veut dire que t_1 et t_2 sont exécutables à partir d'un marquage ?

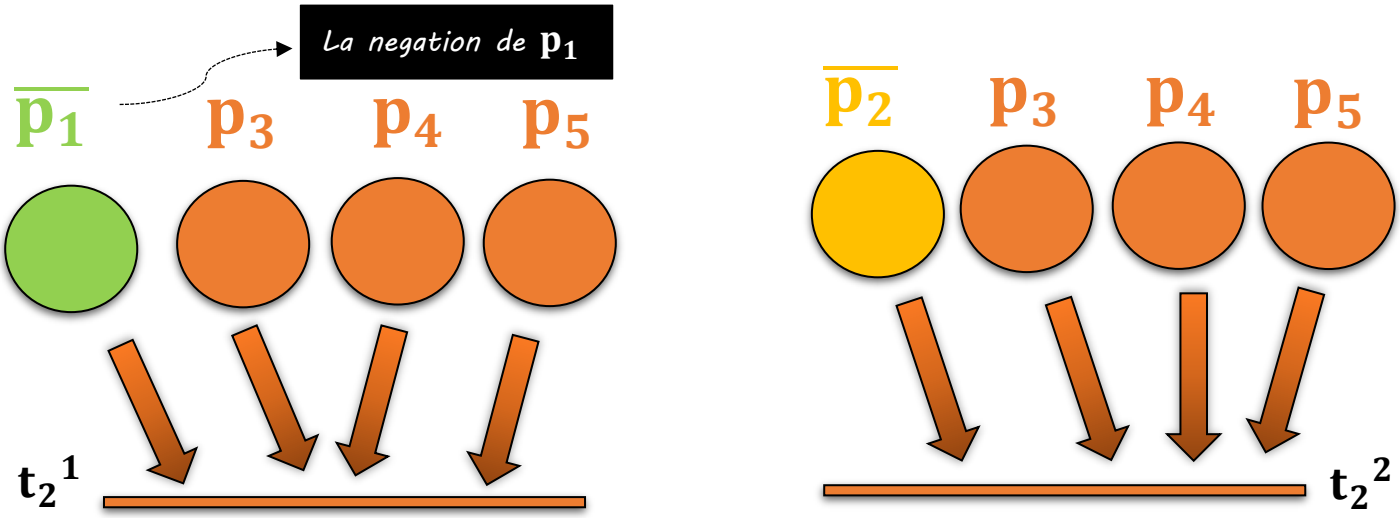


$$t_1 > t_2$$

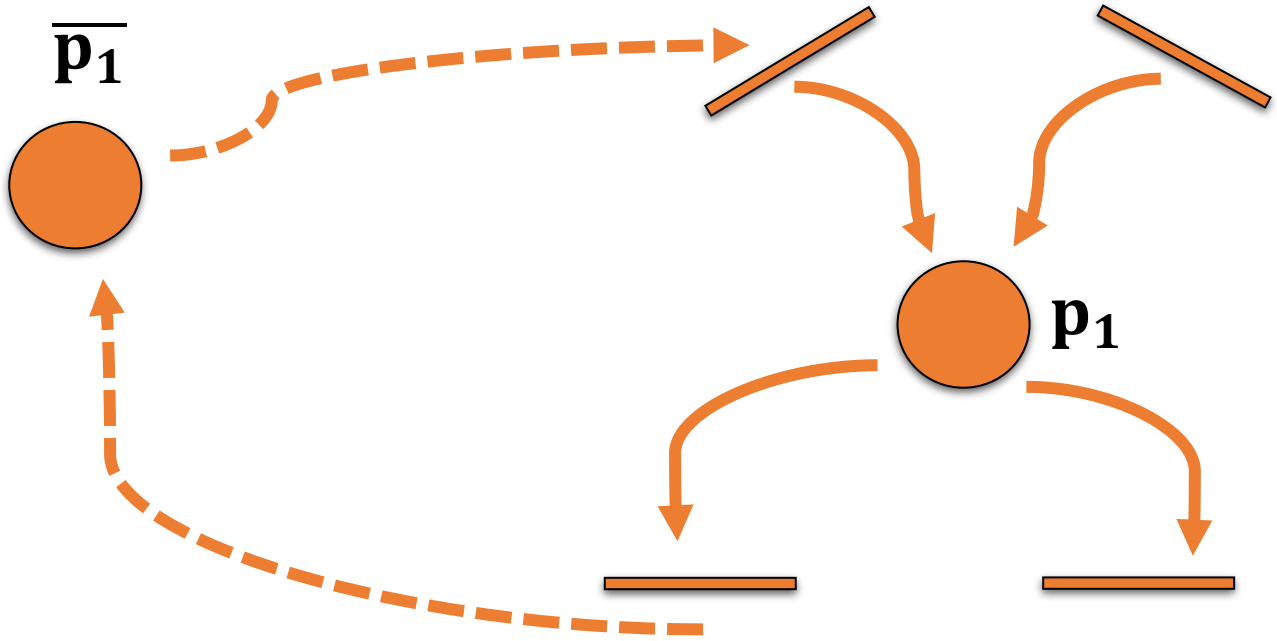
On considère que les places sont disjointes.

Pour exécuter $\mathbf{t_2}$ *il ne suffit pas de dire que* $\mathbf{p_3} \wedge \mathbf{p_4} \wedge \mathbf{p_5}$ *soit vrai, mais aussi que la condition* $\mathbf{t_1}\,(\mathbf{p_1} \wedge \mathbf{p_2})$ *soit fausse.*

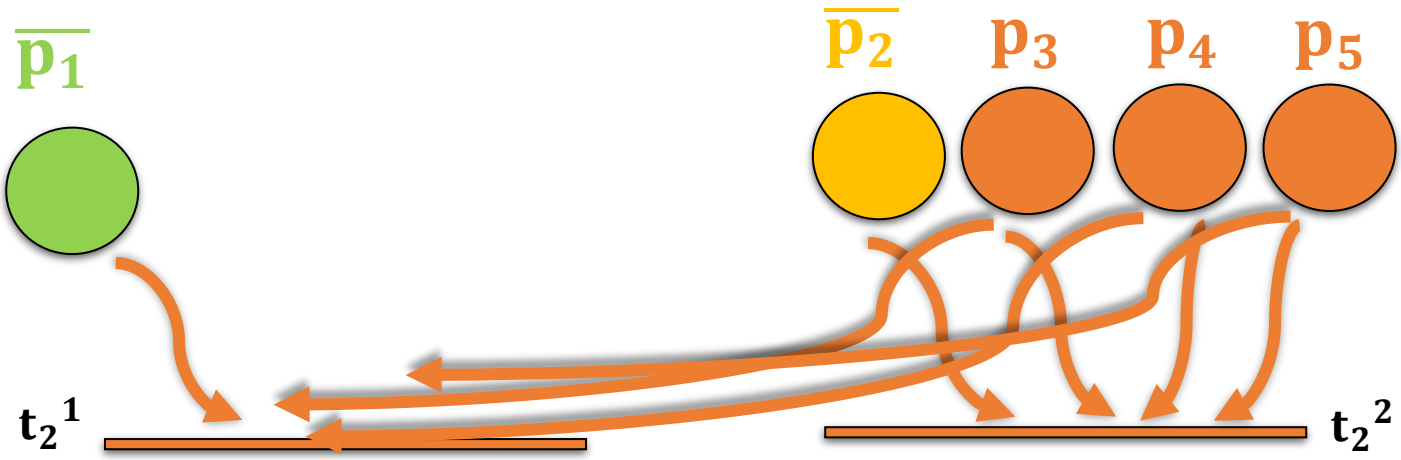
Donc, de manière générale, si j'ai cette priorité $\mathbf{t_1} > \mathbf{t_2}$ *je vais construire un nouveau réseau.*



Pour crée $\overline{\mathbf{p_1}}$



Donc,



Donc, on peut se débrouiller sans les priorités, mais c'est plus agréable de les utilises car les priorités améliorent la lisibilité du modèle.

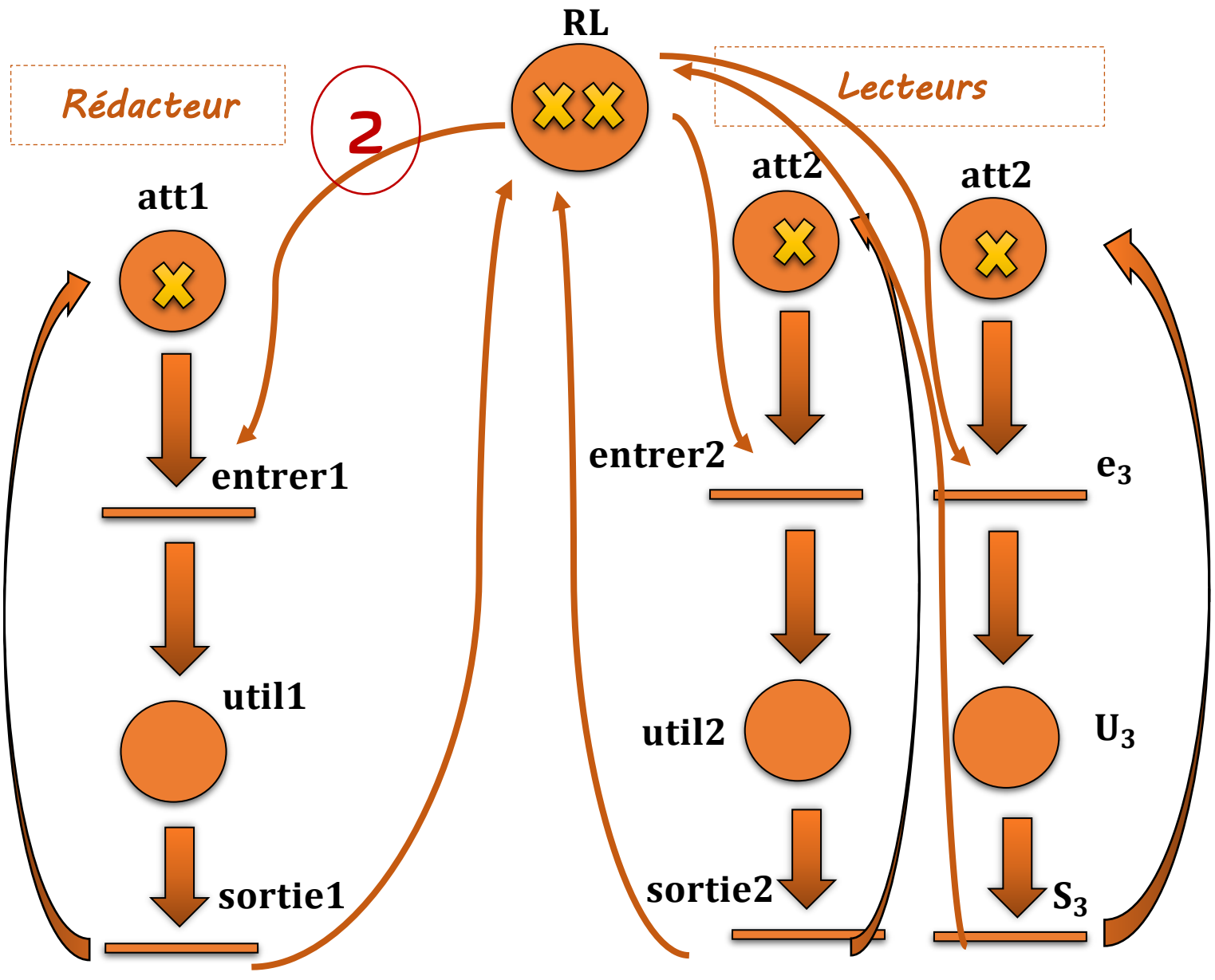
Cet exercice nous a permis de voir les priorités sur le cas borné.

Exercice 4 · Exclusion mutuelle (sans priorités)

Revenons maintenant au scénario vu en CM (cm 3) : plusieurs lecteurs peuvent lire en parallèle VS· un rédacteur, en exclusion mutuelle· Maintenant, supposons qu'on ne sait pas le nombre de lecteurs (on a un nombre de lecteurs arbitraires dans le système)·

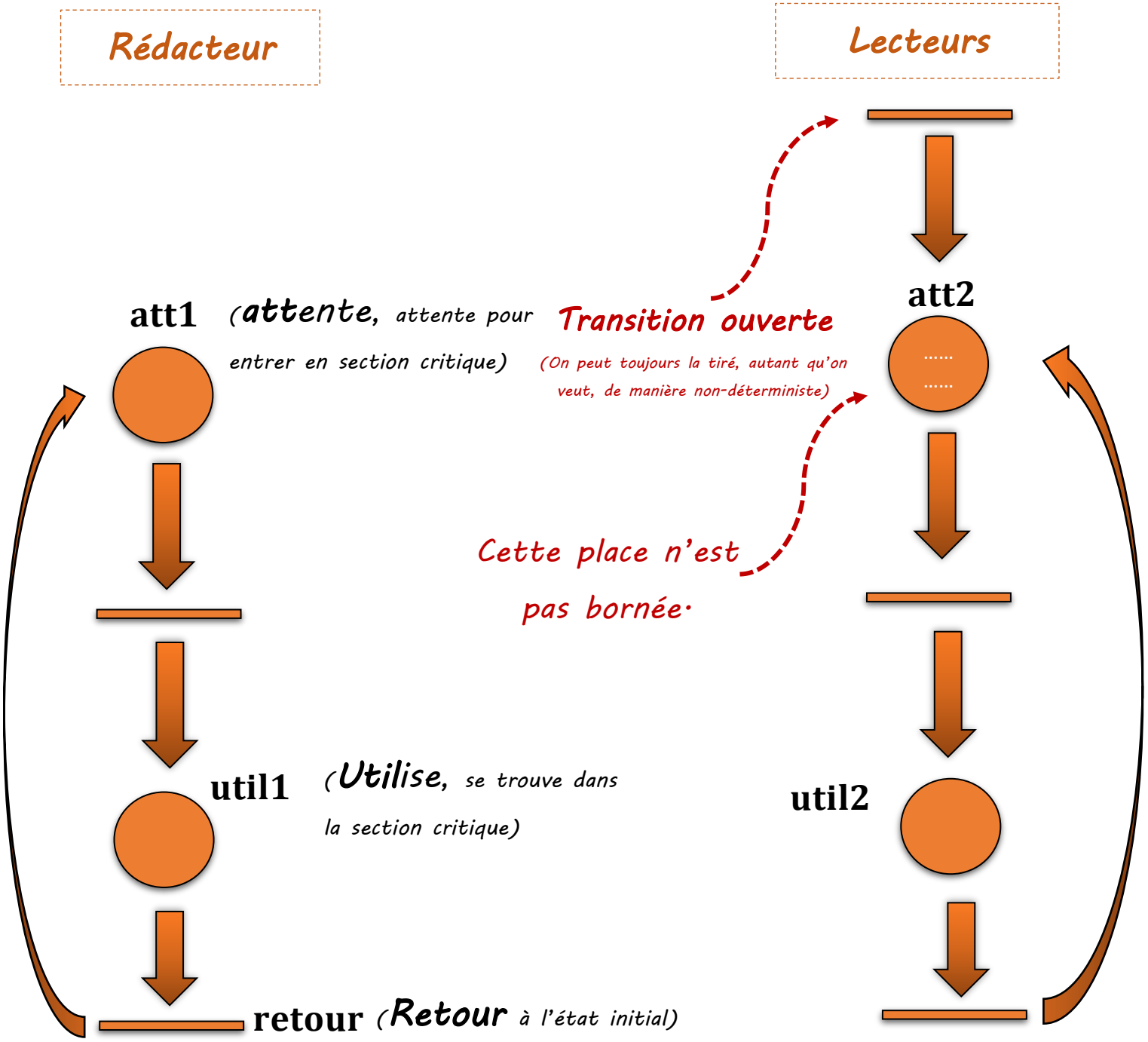
Le scénario vu en CM :

On a 2 lecteurs et 1 rédacteur· Le rédacteur, lorsqu'il écrit, il veut être seul· Cependant, les lecteurs peuvent lire ensemble (pendant ce temp l'écrivain ne peut pas écrire)· Exclusion mutuelle, évidemment·



Le rédacteur a besoin de prendre 2 jetons de RL pour entrer en section critique·

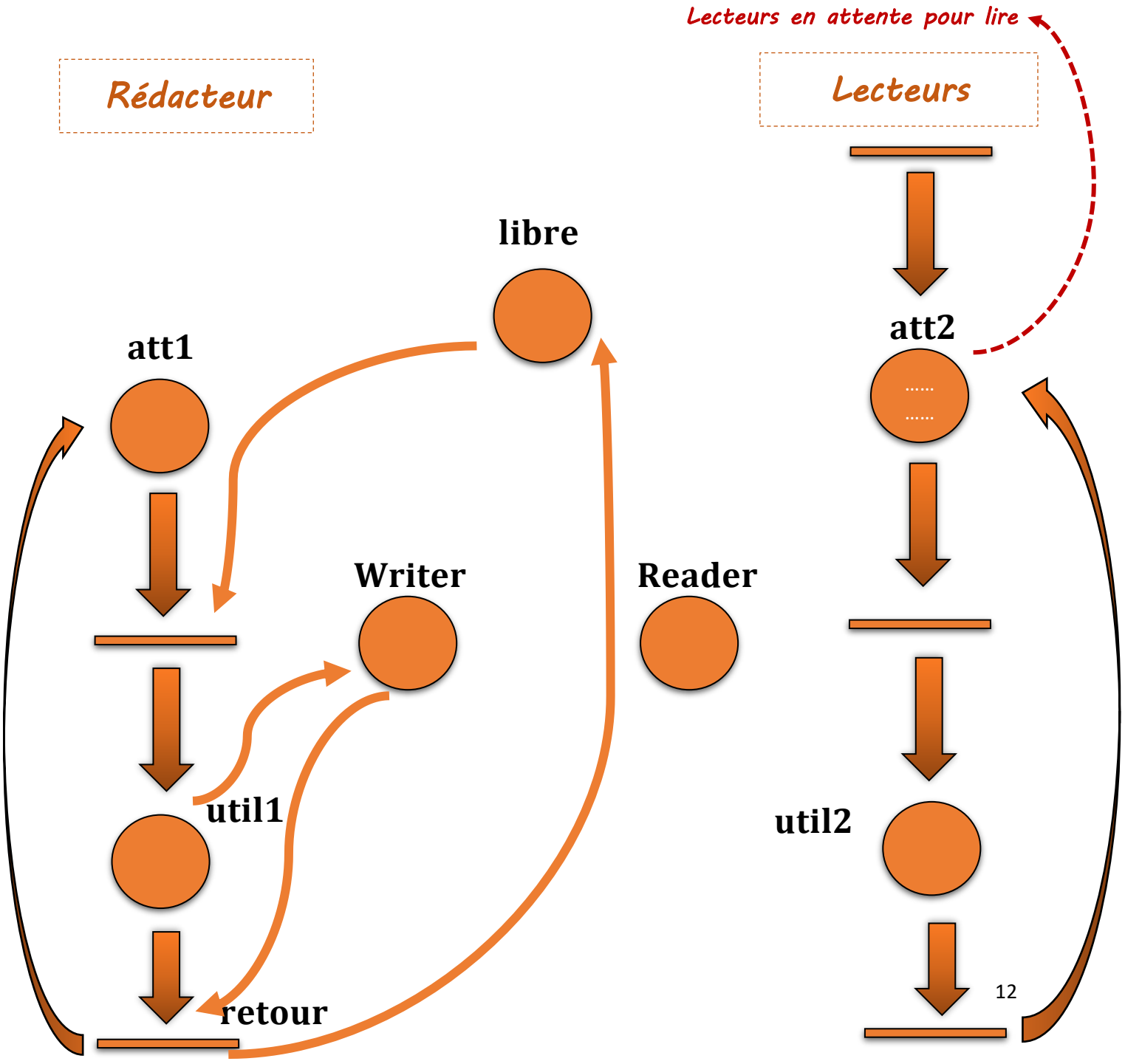
Comment faire pour ce scénario décrit dans cet exo de TD ?



Quelle est la différence entre ce cas-là, et le cas de 2 lecteurs ?

Pour le scénario avec 2 lecteurs, on savait combien de personnes on a (2). Là on ne sait pas, il y a des lecteurs qui arrive tous le temp. Le nombre de jeton est non borné, mais la structure doit rester borné.

Lorsque qu'on a une ressource partagée, faut gères cette ressource. Donc, la ressource est soit libre, soit occupée. Si elle est occupée faut savoir pourquoi elle est occupée (lecture / écriture) et ensuite gérer cette information.



Pour que le premier lecteur qui débarque : faut que la ressource soit libre.

Le 2^{em} : la ressource est utilisée mais en Lecture donc OK.

Comme il y a beaucoup de lecteurs qui attende, ils doivent voir que la ressource est en mode lecture. De plus, au moment ou TLM est sorti faut le déclarer pour que le rédacteur le sache.

Comment savoir que le dernier lecteur est sorti ?

Comment résoudre ce problème avec un réseau de pétri normal, sans priorités ?

Tant qu'il y a des lecteurs dans « la salle », un lecteur qui sort ne peut pas remettre la ressource au statu « Libre ».

- Comment savoir qu'il y a du monde toujours en train de lire ?
- Comment tester si je suis le dernier lecteur à sortir ?

Est-ce que on peut la faire dans les réseaux de Petri ? non, car on ne peut pas tester si une place est égale a 0 (on ne test jamais a 0).

Du coup, la seule manière de le faire est d'utiliser les priorités ou les arcs inhibiteurs.

Suite : Session de TD numéro 3.