

# *TD3 Réseau de Petri*

*Lundi 04.10.2021*

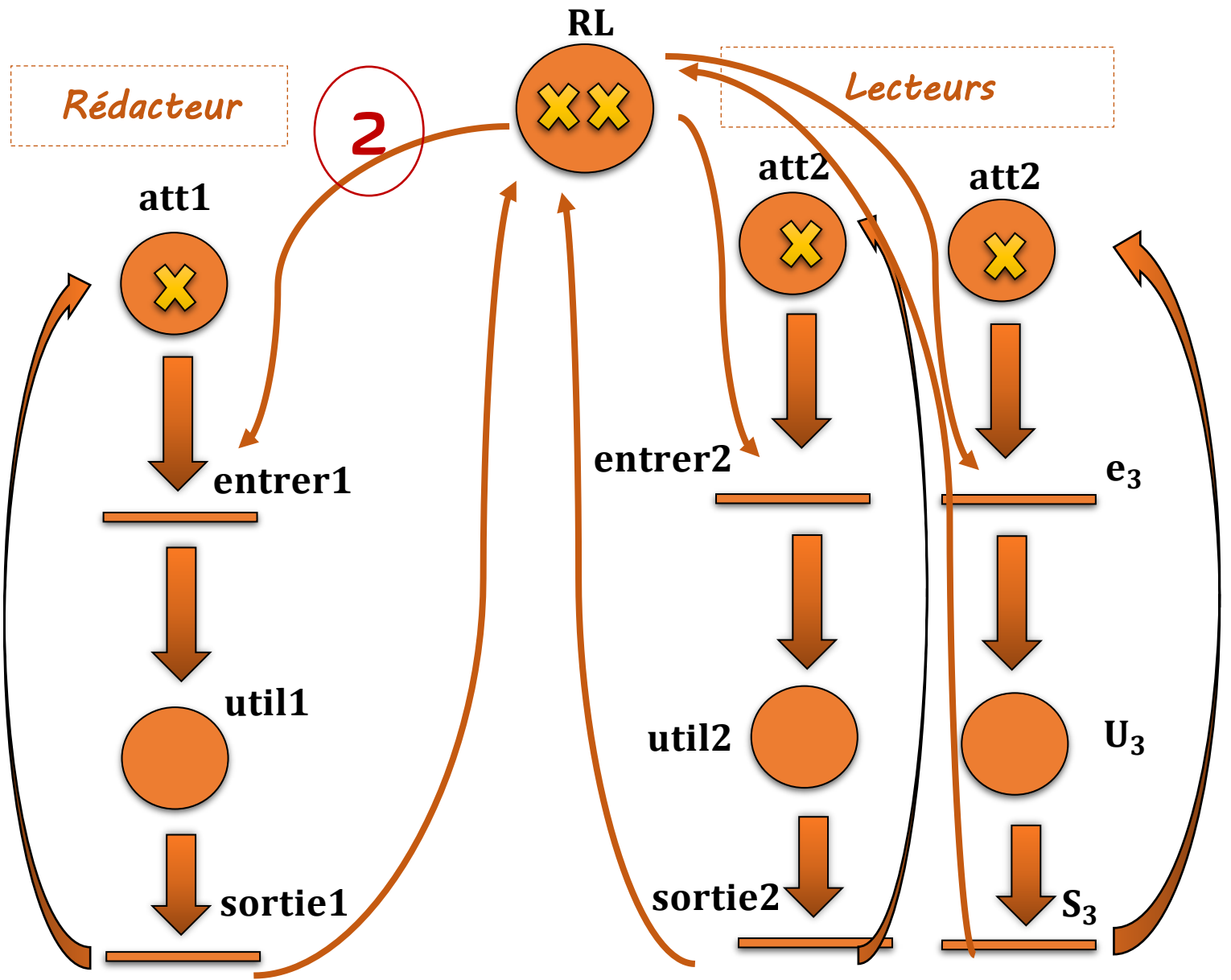
*Fin du TD 2*

## *Exercice 4 • Exclusion mutuelle*

*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 temps 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 ?

Rédacteur

Lecteurs

att1 (*attente*, attente pour  
entrer en section critique)

*Transition ouverte*

(On peut toujours la tirer, autant qu'on  
veut, de manière non-déterministe)

att2

*Cette place n'est  
pas bornée.*

util1 (*Utilise*, se trouve dans  
la section critique)

util2

retour (*Retour* à l'état initial)

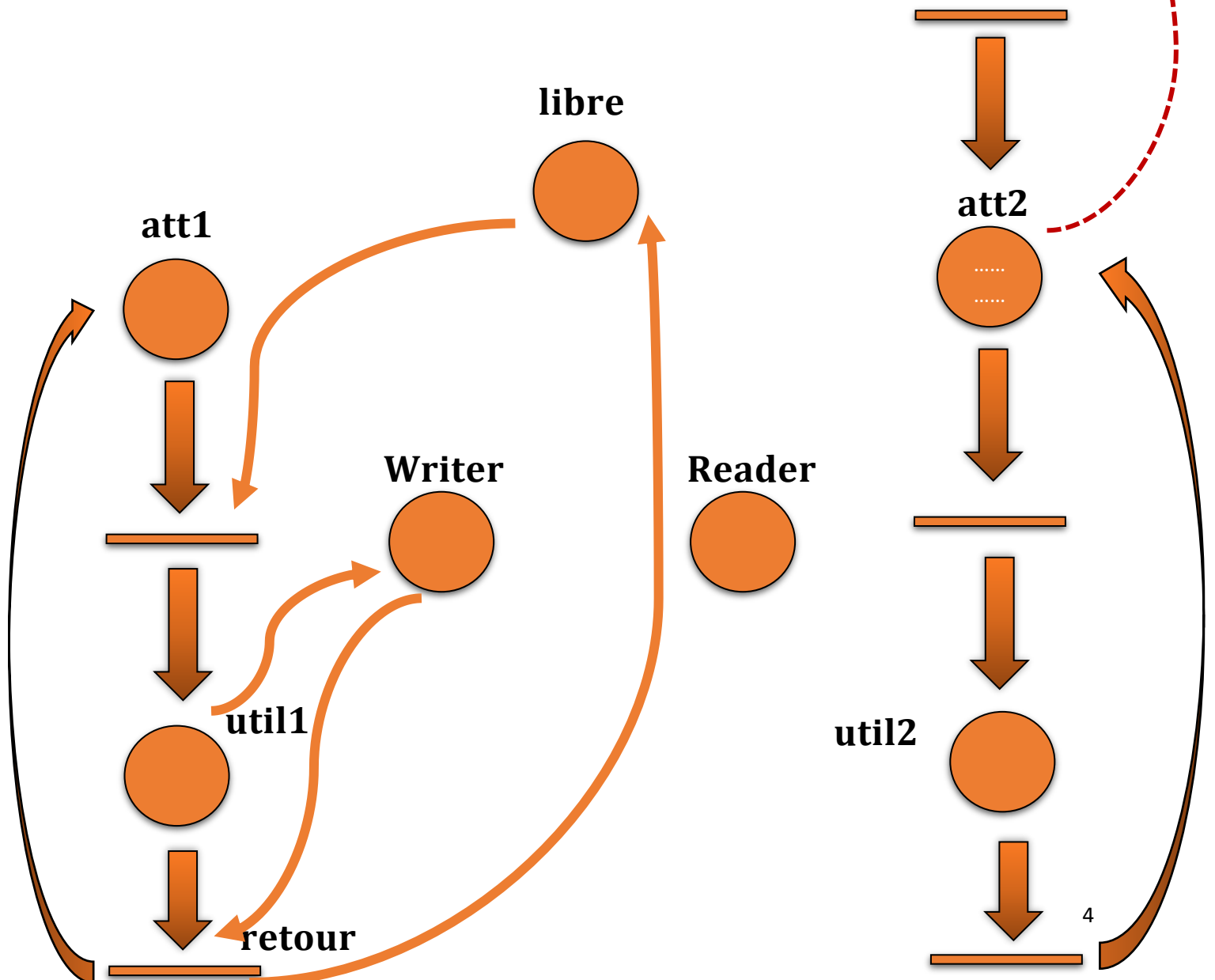
## Quelle est la différence entre ce cas-là, et le cas de 2 lecteurs ?

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 temps. 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.

Rédacteur

Lecteurs



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

*Le 2<sup>em</sup> : 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 ».*

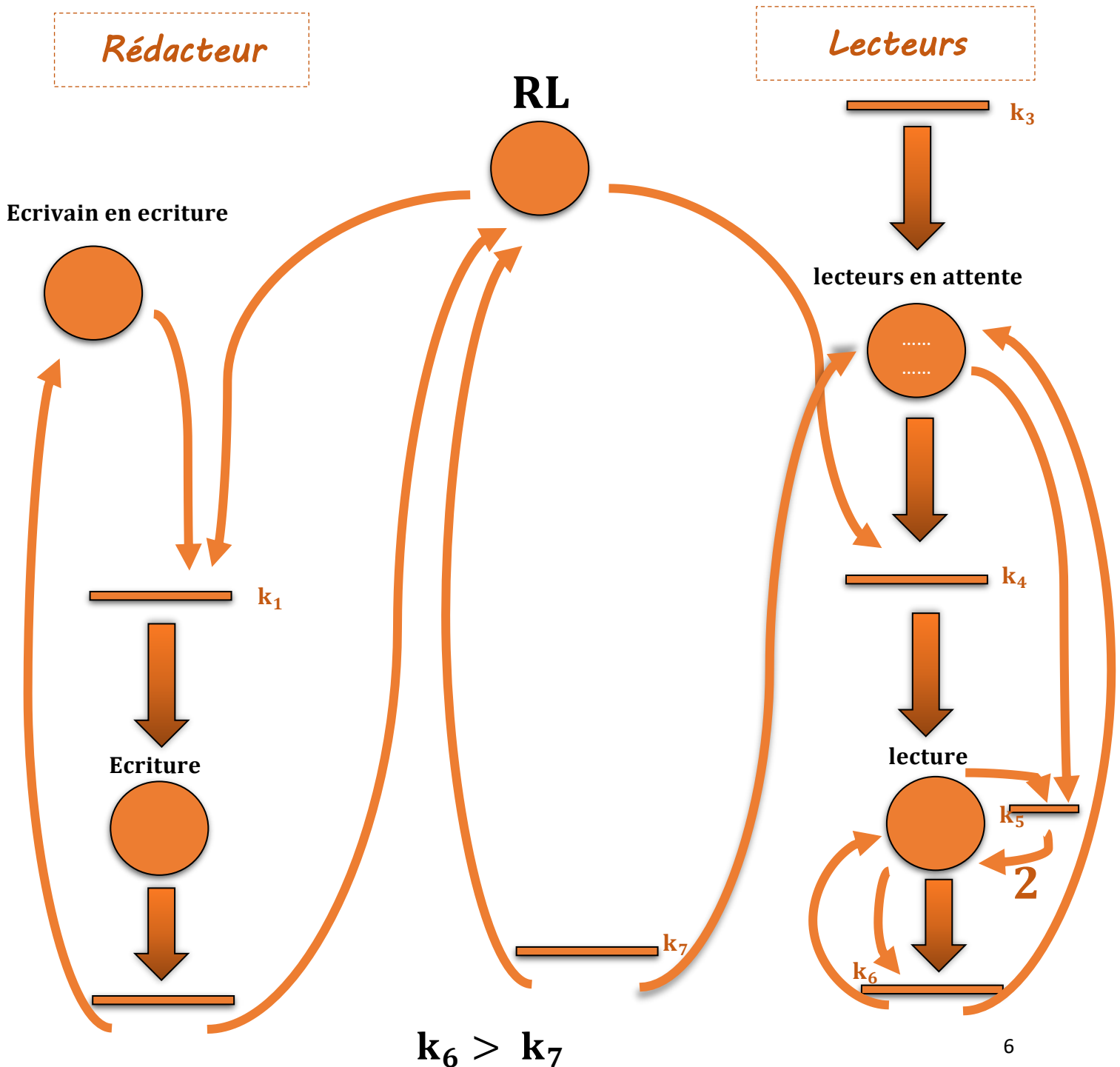
- Comment savoir qu'il y a du monde toujours en train de lire ?*
- Comment tester si je suis le dernier lecteur à sortir ?*

*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).*

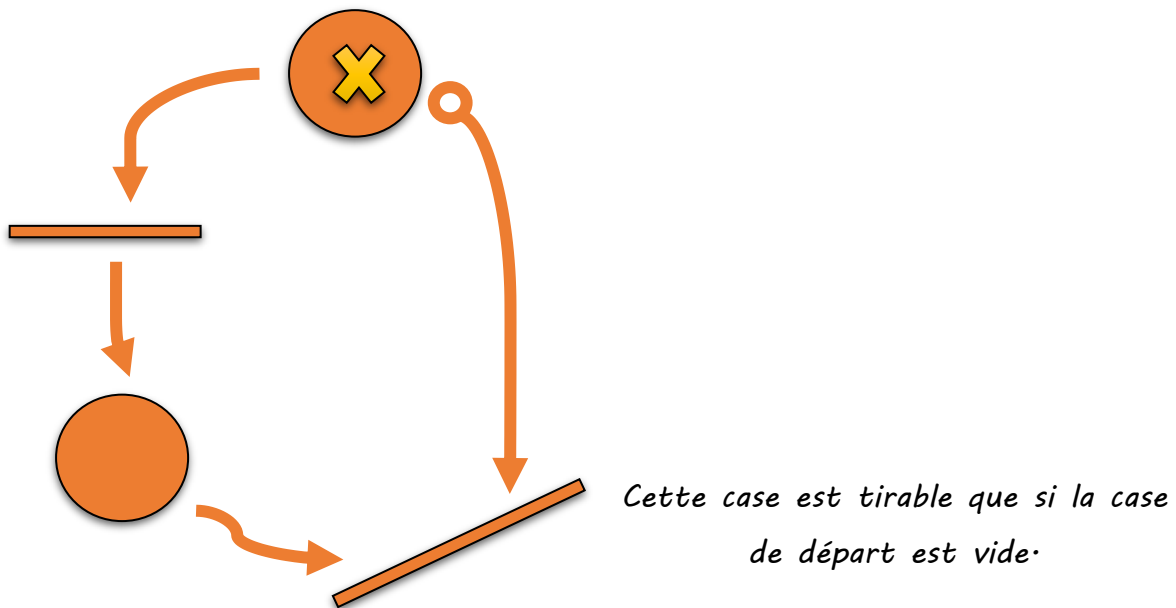
***Du coup, la seule manière de le faire est d'utiliser les priorités ou les arcs inhibiteurs.***

# TD 3

On a un nombre arbitraire de lecteurs qui attende d'entrer. Le rédacteur ne peut pas entrer tant que y'a 1 lecteur qui n'est pas encore sorti. Au départ, lorsque on va donner la lecture, faut savoir si on est en mode lecture ou écriture.



*Tester si une case est égale à 1*



*Maintenant, on va utiliser ça pour trouver une autre solution (sans priorité mais avec arcs inhibiteurs)*

*Rédacteur*

*Lecteurs*

**RL**

**Ecrivain en ecriture**

**lecteurs en attente**

**Ecriture**

**lecture**

$k_1$

$k_3$

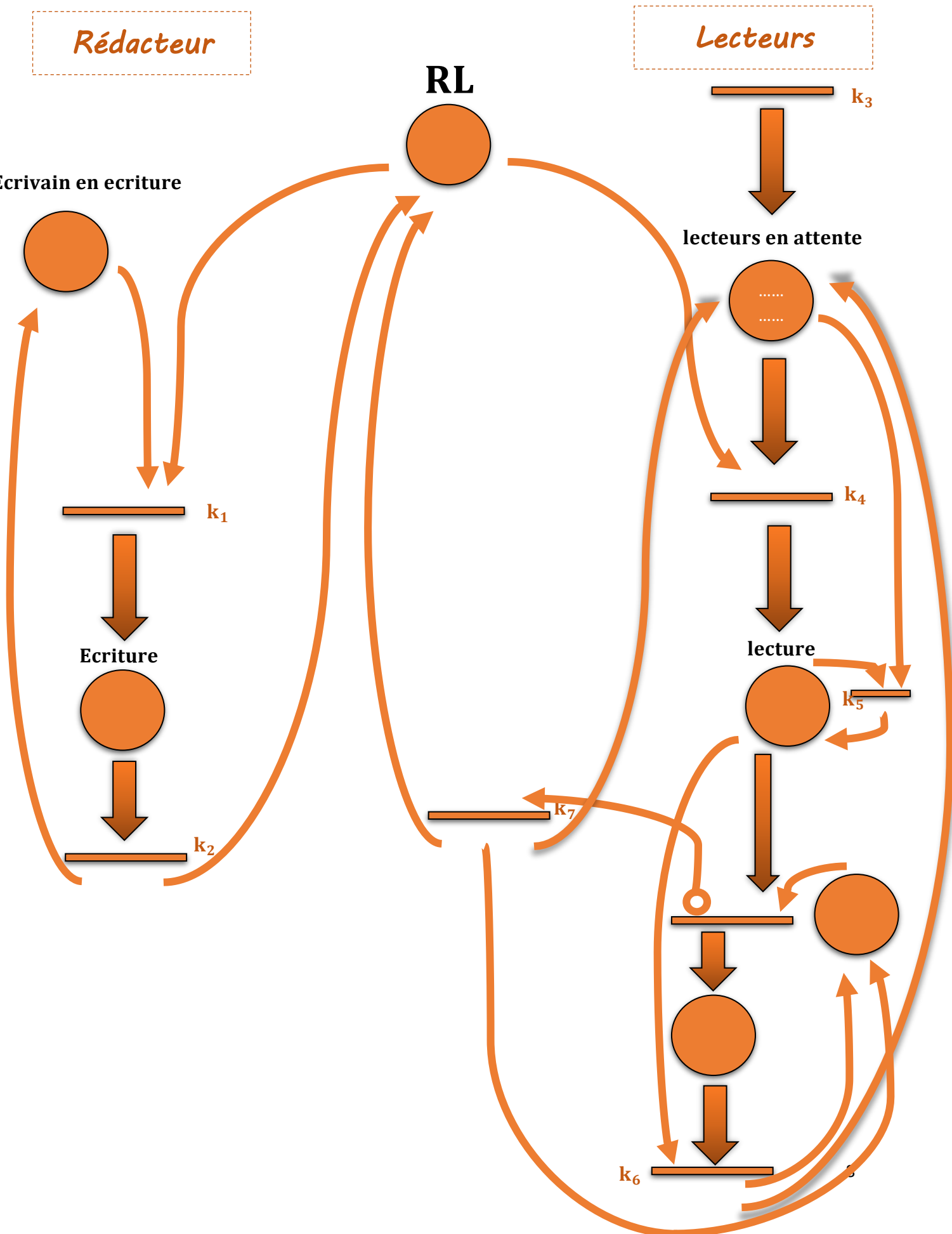
$k_4$

$k_5$

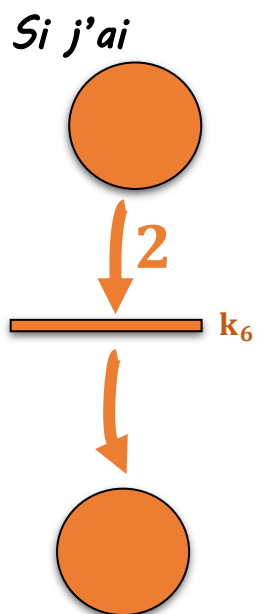
$k_7$

$k_2$

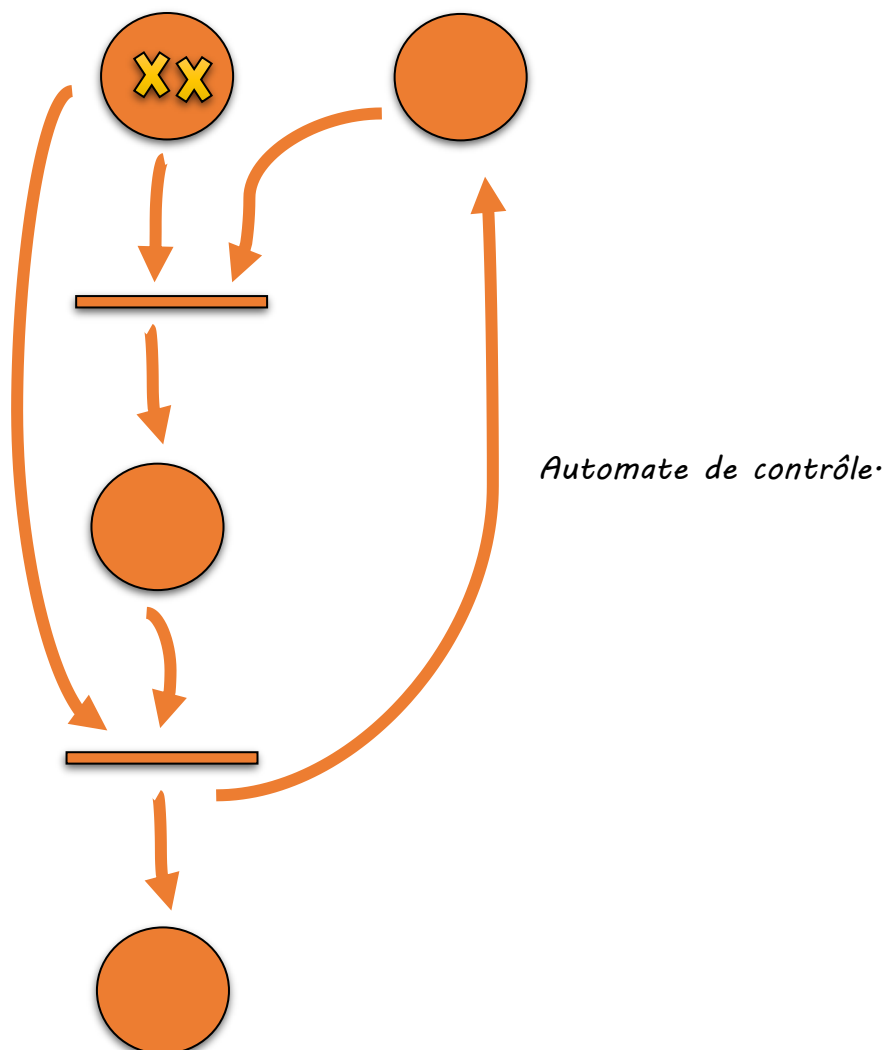
$k_6$





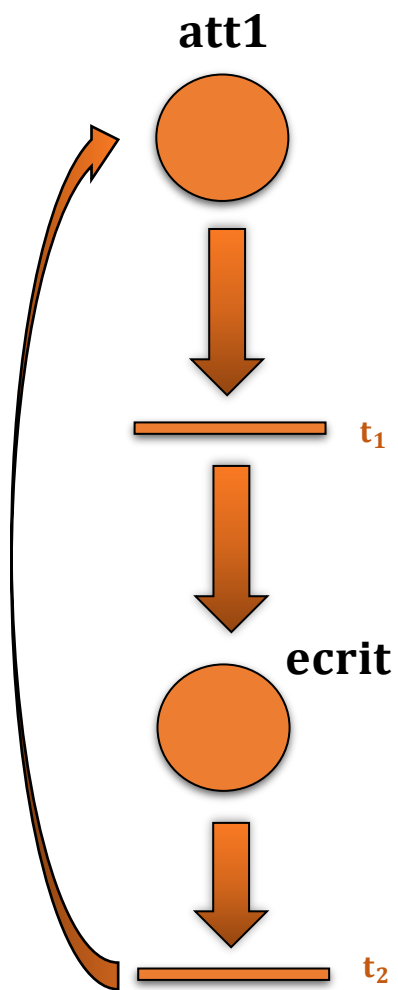


*Je vais le remplacer avec un truc qu'avec des arcs de poids 1.*

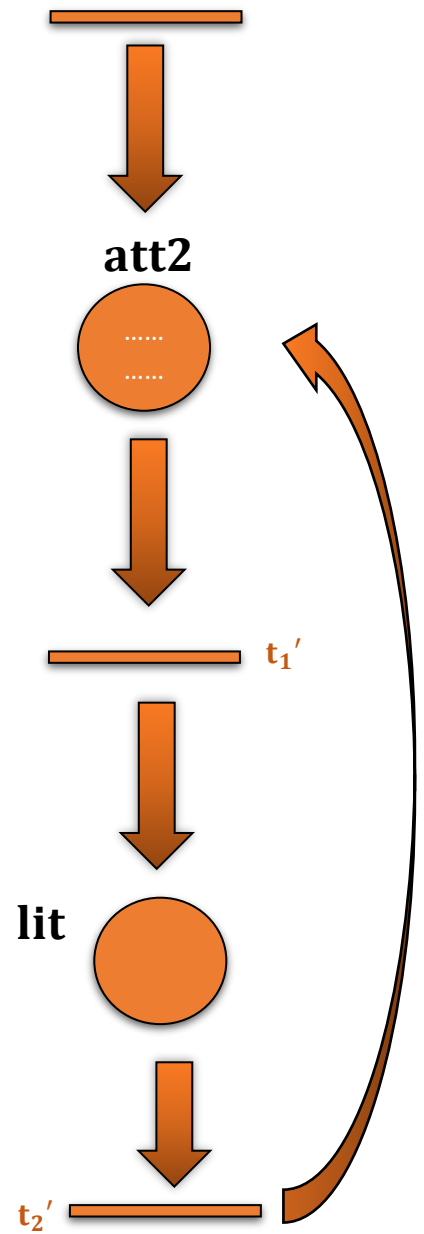


## Autre solution

Rédacteur



Lecteurs



$$\begin{cases} t_2 > t_1' \\ t_2' > t_1 \end{cases}$$