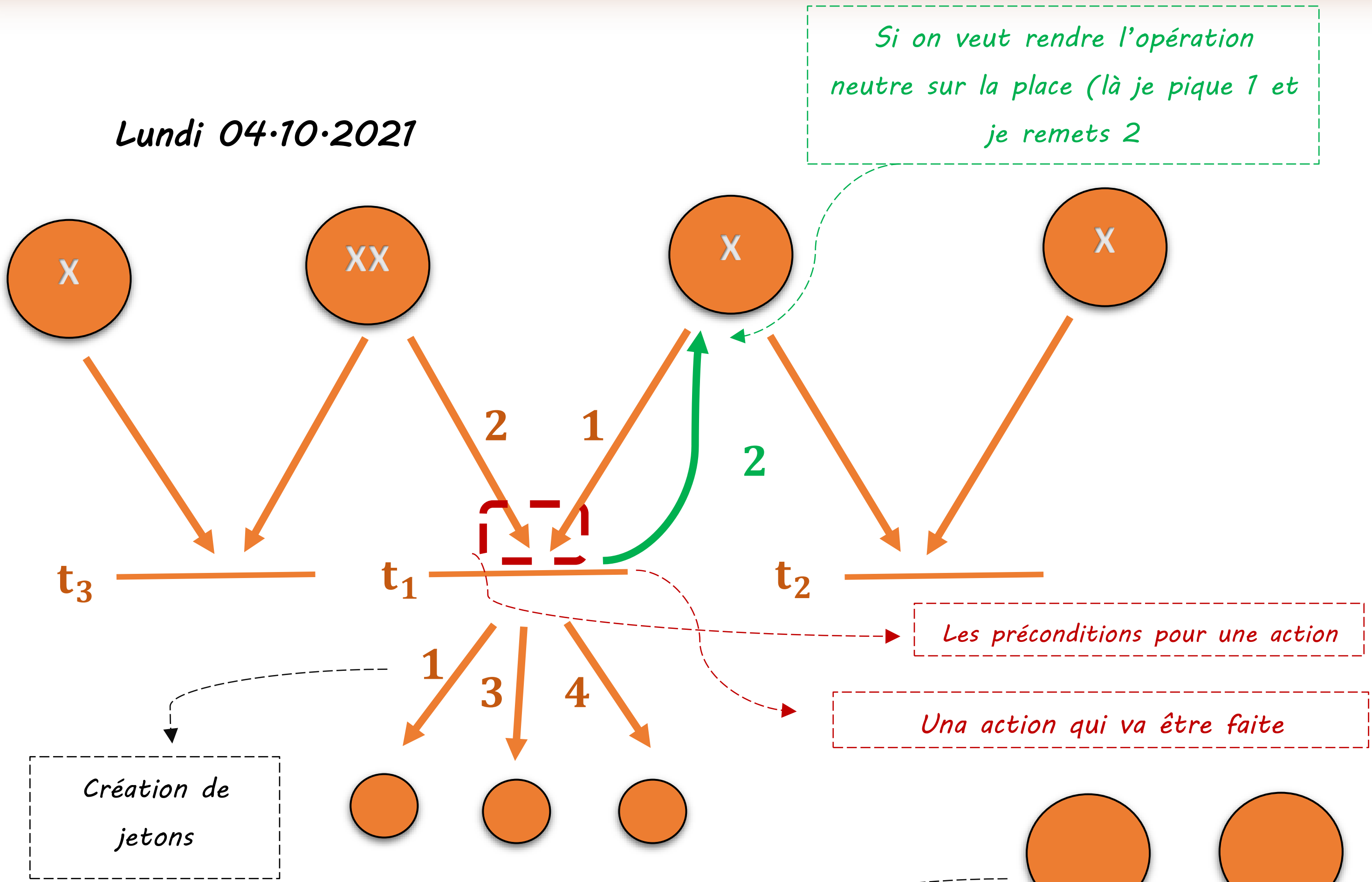
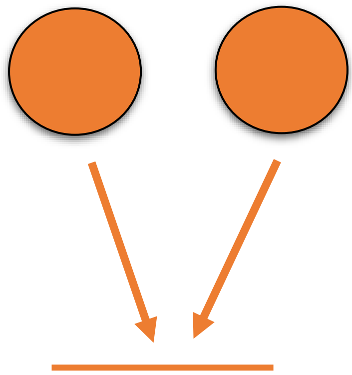
CM4 Réseau de Petri - suite

Lundi 04.10.2021



Dans notre réseau, on peut avoir d'autres places qui n'ont rien à voir :



- Pour chaque action, réfléchir : quelles préconditions ?
- On a vu qu'on peut mettre un certain poids sur les flèches. L'execution de chaque transition fait alors des tests du genre :

$$\boxed{Si\ \ p_1 \geq k, \qquad alors\ p_1 := p_1 - k}$$

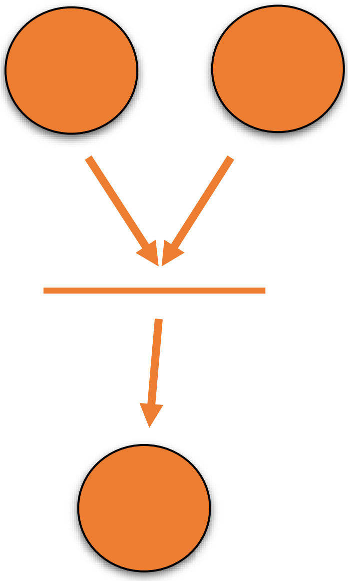
Prend pour valeur

- Et j'ai aussi des operations de la forme :

$$\boxed{p_2 := \ p_2 + k'}$$

- *Chaque une de ces transitions test sur un certain nombre de variables leur valeurs.*
- *On peut tester si certaines variables sont strictement positive, dans ce cas-là je peux les décrémenter :*

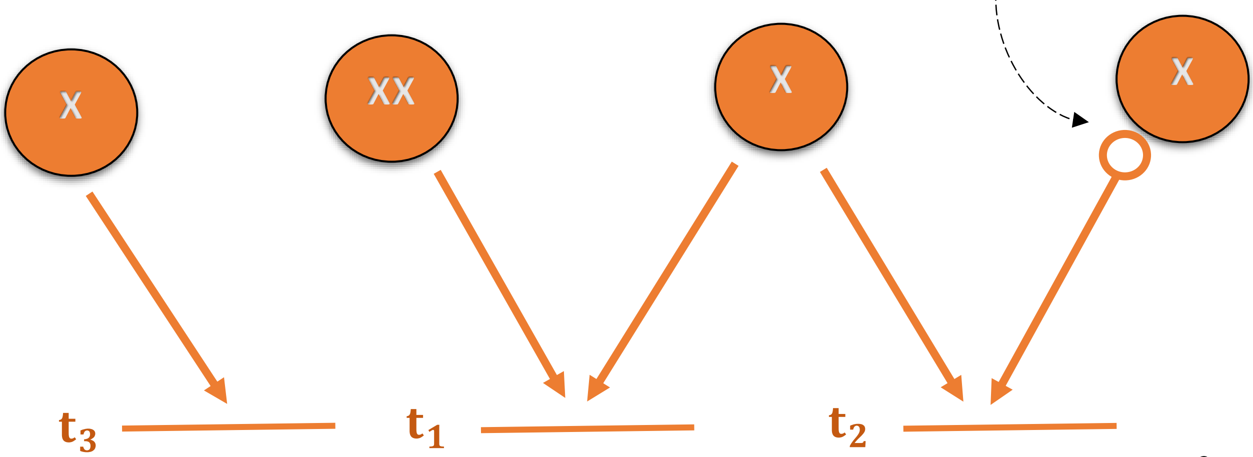
$$\boxed{\begin{array}{c} Si\ \ p_1 \geq 0, \qquad alors\, p_1 := p_1 - 1 \\ p_2 := p_2 + 1 \end{array}}$$



Si ces 2 places sont strictement positives alors je les décrémente

Et là j'incrémente.

- ***Ce qu'on a pas le droit de faire** : on peut jamais dire « Si p = 0 , alors j'incrémente p2 (ou toute autres place) ». Ce n'est pas dans la définition des réseaux de pétri classiques, or il y a des extensions de réseau de Petri qui permettre ça. On va le noter comme ça :*



RdP (Réseau de Petri) avec arc inhibiteur

$$\text{Rdp} + \text{arc inhibiteur} \simeq \left| \begin{array}{l} \text{Machine de Turing} \\ \text{machine a compteurs} \end{array} \right.$$

RdP avec priorité

- *Utile pour résoudre des conflits (donner la priorité à un des participants)*

$$\underline{\gamma} \subseteq \text{T x T} \qquad \text{Relation d'ordre de priorité}$$

$$^{\bullet}t = \{p \in P | (p,t) \in \Delta\}$$

$$^{\bullet}t_I = \{p \in P | (p,t) \in I\}$$

$$M \rhd^t M' \ ssi \ \overbrace{\boxed{\forall\, p \in {}^{\bullet}t_I\, M(p) = 0}}^{Ce\ qui\ change}$$

$$\underbrace{\boxed{\forall\, p \in {}^{\bullet}t\, M(p) \geq 1}}_{On\ garde\ lautre\ condition}$$

Comment définir la priorité ?

Celui qui choisi entre les transitions, lorsque tt les 2 (t1, t2) sont exécutable à partir d'un certain marquage, si on lui dit qu'une transition doit passer avant l'autre, alors l'autre peut s'exécuter que quand la 1ère ne peut pas s'exécuter.

Comment ça se définit formellement ?

Exécution avec priorité

Si c'est possible d'exécuter M pour aller à M'

$$M \underset{\underline{\gamma}}{\rhd^{t}} M' \quad ssi \quad M \rhd^{t} M'$$

$$et \ \forall t' \in T, \quad \exists M'' \quad M \rhd^{t'} M''$$

$$\Longrightarrow$$

$$\neg(t' > t)$$

RdP avec arc inhibiteur et RdP avec priorité

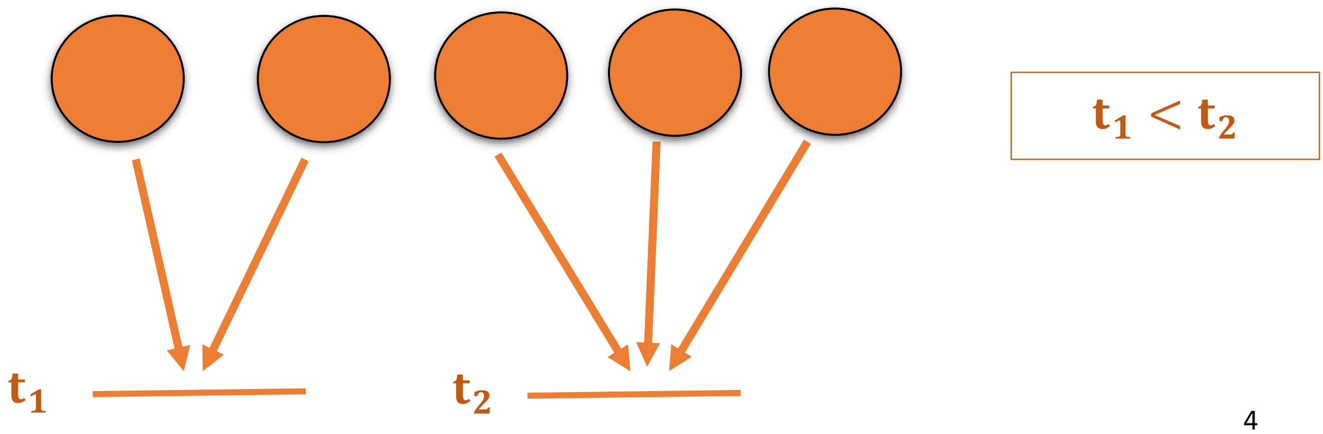
La question qui se pose : ces systèmes avec priorité, je peux les coder avec les systèmes avec arc inhibiteur (et inversement) ?

La semaine dernière, on a vu dans les exos que pour les « 1-safe » (1-borné), les priorités c'est que du sucre syntaxique. Mais de manière générale ? Si on nous donne les arcs inhibiteurs on peut coder les priorités ? Et l'inverse ?

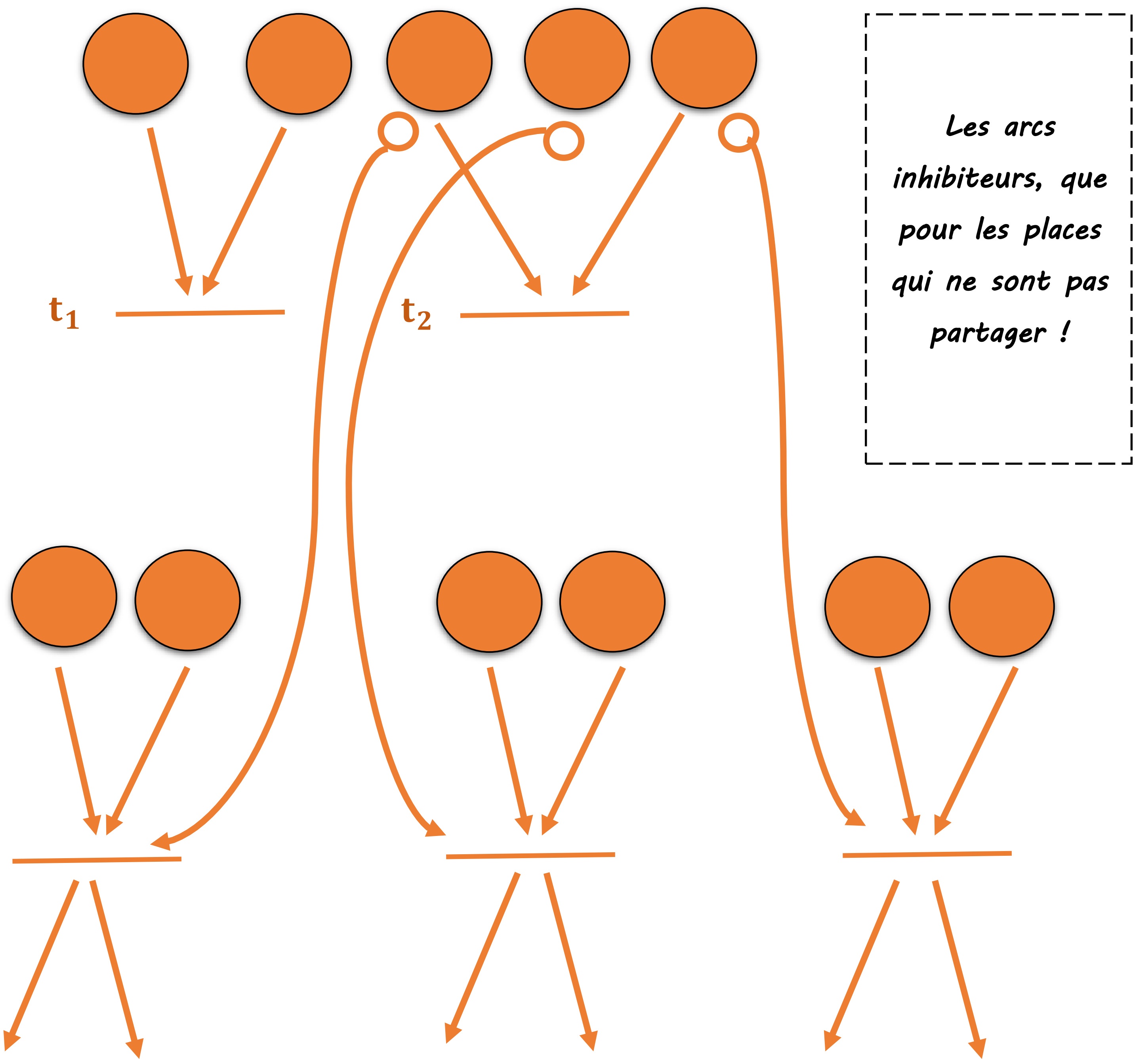
Si j'ai des arcs inhibiteur (possibilité de tester a 0), je peux coder les priorités ?

Prenons un exemple :

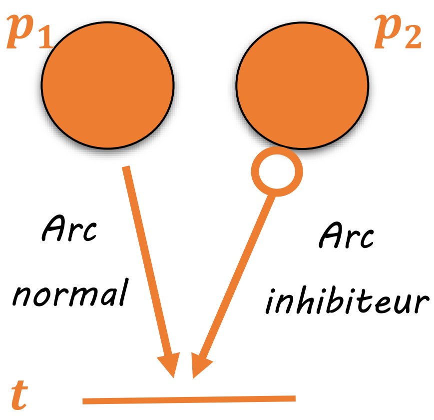
On a une transition t1 et une transition t2, chaque une a un certain nombre de conditions et on dit que t1 < t2. Donc, comment coder ça ?



Lorsque on a la priorité, ça veut dire que si on peut exécuter t1 et t2, alors on exécute t2. Donc, pour t2 il suffit qu'il remplît ses préconditions et il passe. Pour t1, il doit montrer qu'il remplît ses préconditions et que t2 ne peut pas s'exécuter (ça veut dire qu'au moins une des places avant t2 est vide). Donc, on peut remplacer ce réseau par un autre

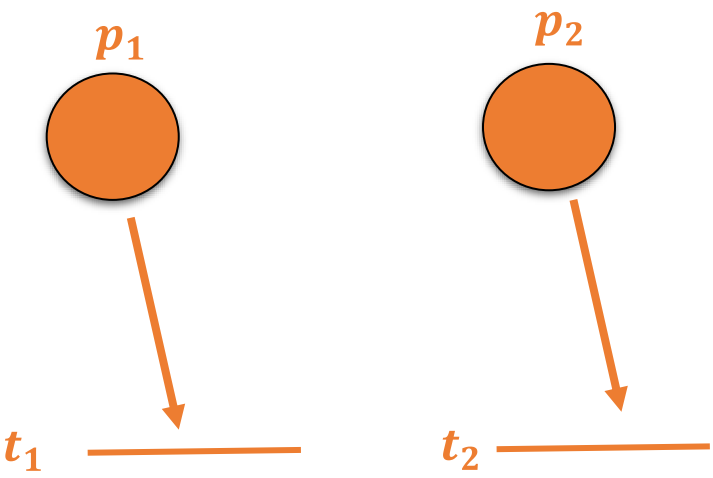


Mtn, les priorités sont aussi puissantes pour coder les arcs inhibiteurs ? (toujours dans le cas non-borné). Prenons un exemple simple :



Si j'ai 1 a P1 et 0 a P2, c'est qu'alors qu'on peut exécuter ! Comment simuler cette chose avec les priorités ?

Réponse :



$$t_2 > t_1$$

On a donc le thm (théorème) des inversement !

C'est très utile en pratique.

On peut analyser, une fois qu'on a réalisé un modèle, si c'est correct ou faux ? Donc on va construire le graphe de marquage accessible, ce que normalement est possible que lorsque c'est borné, mais on va voir quand même qu'on peut s'en sortir.

Graphe de marquage

*Si j'ai un réseau de Petri **R**,*

$$R = (P, T, \Delta)$$

A partir d'un certain M_{init}

$$G(R) = \Big(\ \overbrace{[P \to \mathbb{N}]}^{\textit{Ensemble de marquages}}\ ,\ \overbrace{\Delta}^{\textit{relation de transition entre marquages}}\ \Big)$$

$$Acc(M_{init}, R) = \textit{ ensemble des marquages } M \textit{ t.q. } M_{init} \rhd^{*} M$$

Comment résoudre ce pbm ?

Pbm d'accessibilité :

Etant donné M dire si $Acc(M_{init}, R)$,
Si on me donne un chemin pour y aller, il y a une erreur dans le system.

Si je suis en réseau borné, c'est comme la question d'accessibilité dans un graphe.

- **Cas borné :** *en théorie on peut faire un parcours de graphe, mais la complexité est énorme ! Par exemple, un réseau de Petri avec n places ? quelle est la taille du graphe de marquage (borné-1) ? 2^n (n: nb de places ; 2 : nb de valeurs possibles). Donc, ça peut être exponentielle en fonction de la taille du réseau. Donc, prendre le graphe et le parcourir ce n'est pas aisé, donc faut avoir des algos plus intelligents.*
- **Cas non-borné :** *Le graphe est infini, existe-il quand même des algos qui permette d'analyser ? Ce qu'on cherche c'est de résoudre un autre pbm aussi complexe, mais pour ce pbm il y a un algo qui nous donne la solution : problème de couverture.*

Problème de couverture

<u>*On va définir la relation d'ordre sur les marquages :*</u>

Soit $\leq$ *la relation d'ordre naturelle sur les marquages :*

$$(n_1, \dots, n_k) \leq (n_1', \dots, n_k') \qquad ssi \;\; \forall\, i, \qquad 1 \leq i \leq k \quad n_i \leq n_i'$$

Mtn, je ne vais pas me poser la question si je vais atteindre M, mais s'il y a quelque chose de plus grand qui est atteignable ?

$$\boxed{Etant\ donné\ \ M :\ \exists\, \boldsymbol{M'} \geq \boldsymbol{M}\ \ t.q.\ M' \in\ Acc(M_{init}, R)\, ?}$$

Donc, on cherche un état qui viole l'exclusion mutuelle, si on trouve un état plus grand, c'est même encore pire, car il contient cette violation de l'exclusion mutuelle.

Lorsque on pose la question $\exists\, \boldsymbol{M'} \geq \boldsymbol{M}$ *il y a des algos qui sont applicable.*

Quelle est l'intuition mtn ? on va construire un arbre, partir d'un certain marquage et on va regarder qui sont les successeurs possibles, et ainsi de suite (profondeur ou largeur d'abord). Evidemment, l'arbre est infini, et c'est là qu'on a une petite idée. Supposons qu'on trouve un certain marquage :

$$(n_1{}^{\circ}, \dots, n_k{}^{\circ})$$

$$(n_1{}^{j}, \dots, n_k{}^{j}) \qquad (\dots) \qquad (\dots) \qquad (\dots)$$

$$\geq$$

$$(n_1{}^{i}, \dots, n_k{}^{i})$$

ω Oméga

On arrive vers un certain contenu qui est plus grand !
Donc, à partir de ce marquage on voit qu'il y a plus de jetons (si on nous donne un marquage et on nous donne des jetons on peut refaire l'action, mais si on le refait, on va retrouver quelque chose de plus grand, ce qui veut dire que cette place ou le nombre a augmenté est non-borné, donc on va augmenter le nombre de jetons. On va le remplacer par un symbole spécial, oméga, on va abstraire tous les nb extérieurs par oméga (oméga est en puit san fon)

Question

<u>Mais qui nous garantit qu'on va rencontrer une telle situation ?</u>
Si j'ai un chemin qui va vers l'infini et je ne tombe jamais sur un truc supérieur :

<u>1-borné</u>
Si on part de 6 et on n'a pas le droit au supérieur, faut descendre, et plus on descend, plus notre situation s'aggrave.

$$6 \ldots\ldots\ldots \overbrace{i \ldots\ldots\ldots j}^{i<j}$$

$$(6,5) \qquad (4,7) \quad \underbrace{(5,4)}_{ou\ (3,10)}$$

Je suis borné, du coup faut que je monte.

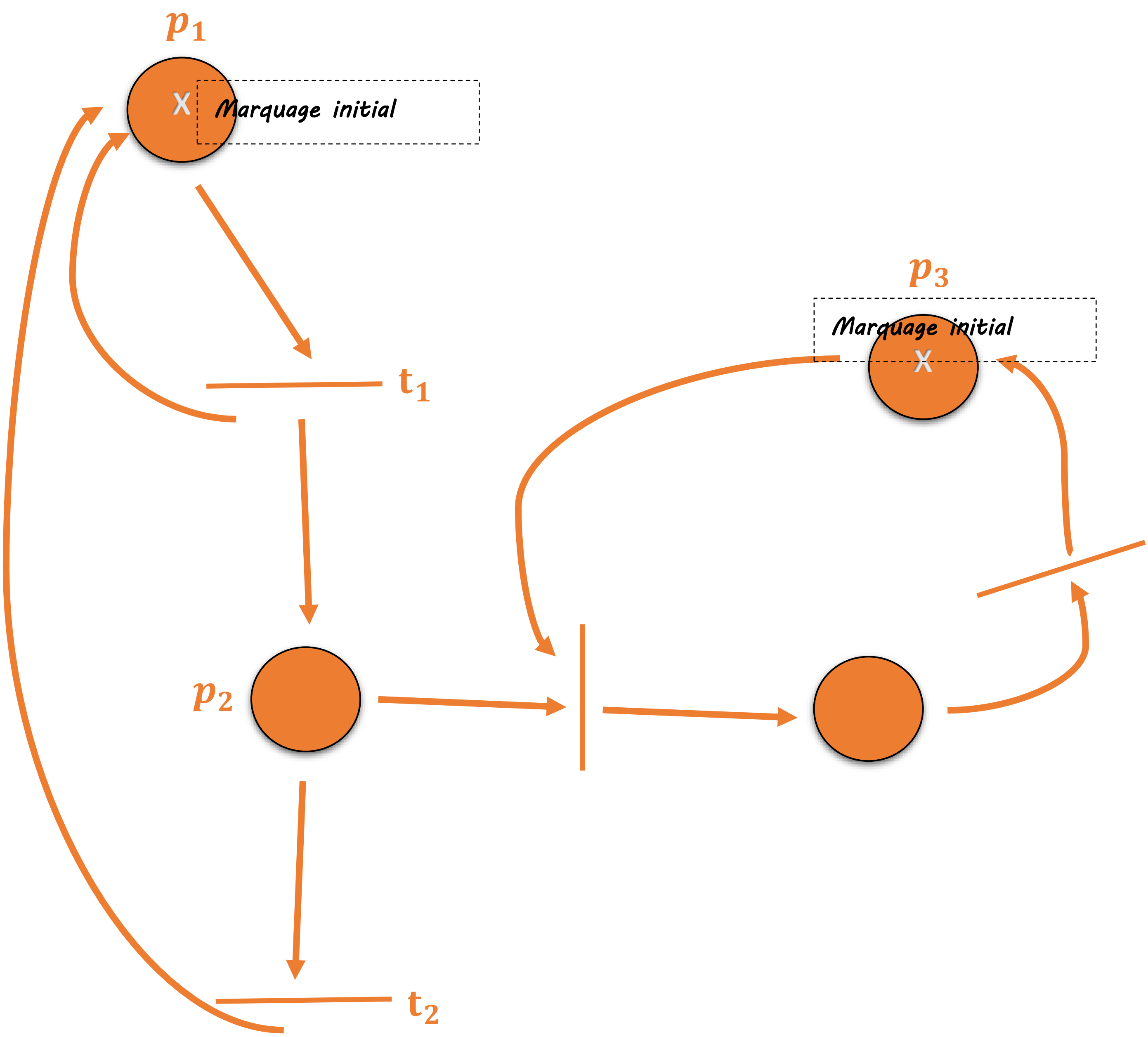
Thm

La relation $\leq$ a une certaine propriété intéressante, elle contient 2 points tel que l'un est inférieur ou égal à l'autre (car on est borné en bas avec le 0).

On peut démontrerai que sur n'importe quelle séquence infinie, on aura au moins 2 points comparables, et dit qu'il y a une comparaison, 2 possibilités :

- *Je revoie le même -> boucle infini -> je m'arrête.*
- *J'ai une dimension qui est la même -> je pose un oméga, plus on va mettre des omégas, plus on va converger.*

Ce thm nous garantit l'arrêt de cet algo. On va voir ça sur un exemple :



Notre marquage initial

$$(1,0,1,0)$$

$$\downarrow t_1$$

On constate que même avant de continuer à développer on voit qu'il suffit de répéter t1, on voit des jetons dans t2 donc on remplace par oméga

$\geq$

$$(1,\overbrace{\cancel{1}}^{\omega},1,0)$$

Je peux puiser dans oméga comme je veux.

$\geq$

$$\xrightarrow{t_1}\ (1,\overbrace{\cancel{2}}^{\omega},1,0) \qquad \xrightarrow{t_2}\ (\overbrace{\cancel{2}}^{\omega},\omega,1,0) \qquad \xrightarrow{t_3}\ (1,\overbrace{\cancel{0}}^{\omega},0,0)$$

Là c'est bon, c'est clair que je vais tourner en rond

Là on va s'arrêter car la boucle va puiser dans l'oméga donc on remplace par oméga

$$(1,\overbrace{\cancel{0}}^{\omega},0,0) \xrightarrow{t_4} (1,\omega,1,0)$$

On a déjà vu ce truc donc on s'arrête.

Que nous donne ce graphe ? Ou se trouve les places non-borné ? Là où on a trouvé oméga. Donc, les places 1 et 2 sont non-borné, et il y a   qui sont borné par 1.

2em information : comme le graphe représente des ensembles, si on a M alors il existe un M' qui couvre ce M et qui appartient au accessible (= M se trouve dans un des ensemble ?)

$$\boxed{M} \quad \exists M' \geq M \qquad M' \in Acc$$

Chaque un des sommets du graphe représente un ensemble de marquage. Résoudre ce pbm revient à tester si $M \in m$ des ensembles de marquages définis par les nœuds de cet arbre. On l'appelle ***l'arbre de karp-Miller (KM)***

C'est utile car la construction de ce graphe est relativement facile (l'arrêt est garanti par le théorème), donc tous les branches sont finies. Cependant, la complexité, en théorie, est horrible car le point d'arrêt, ça risqué de prendre du temp à le trouver.