TD 9: Loi de Little et réseaux de Pétri

lionel.rieg@ens-lyon.fr

Exercice 1 (Loi de Little)

On se propose de démontrer ici la loi de Little : $\mathbb{E}(N) = \lambda\,\mathbb{E}(W)$. Pour cela, on va tout d'abord se placer dans un cadre déterministe et étudier une trajectoire. On se donne un système muni de processus d'arrivée et de départ, dont on note respectivement $A(t)$ et $D(t)$ les processus cumulés. Le nombre de clients dans le système à l'instant t est donc $C(t) = A(t) - D(t)$.

1. Exprimer l'aire entre les courbes de $A(t)$ et de $D(t)$ de deux façons différentes.
2. Définir
   - le taux moyen d'arrivée $\lambda(t)$ dans le système jusqu'à l'instant t,
   - l'attente moyenne $W(t)$ des clients jusqu'à l'instant t,
   - le nombre moyen $N(t)$ de clients dans le système jusqu'à l'instant t.
3. En divisant l'égalité de la première question par t et en faisant tendre t vers $+\infty$, donner la loi de Little expérimentale.
4. On conclut alors en utilisant le théorème ergodique. Sous quelles hypothèses la loi de Little s'applique-t-elle ?

Réseaux de Pétri

Définition (Réseau de Pétri)

Un *réseau de Pétri* est la donnée d'un graphe orienté biparti $(\mathcal{P}, \mathcal{T}, E)$ et d'une fonction $\mu : \mathcal{P} \to \mathbb{N}$.

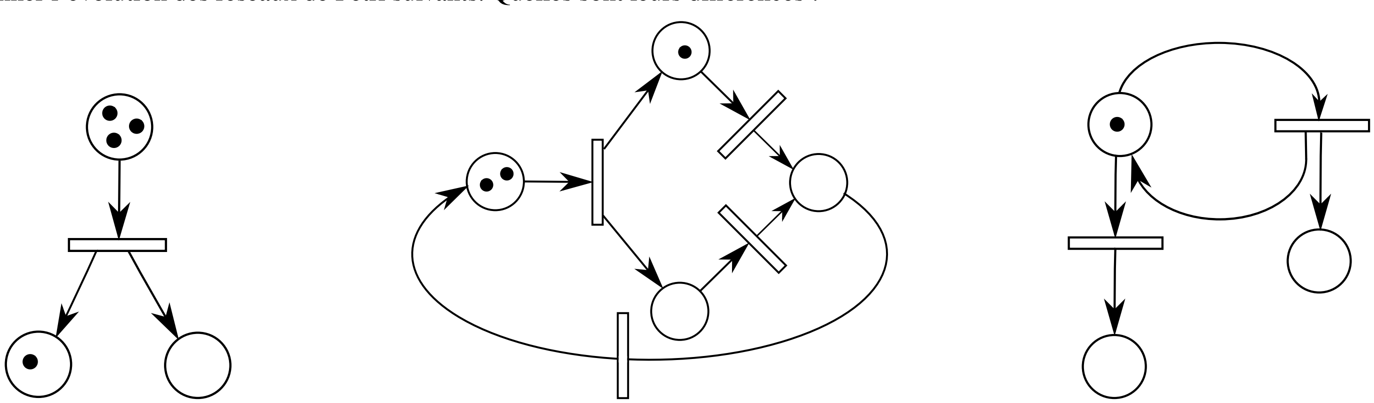
Les éléments de $\mathcal{P}$ sont appelés les *places* et ceux de $\mathcal{T}$ les *transitions*. On note ${}^\bullet t$ et $t^\bullet$ respectivement les voisinages négatifs et positifs d'une transition. On définit de manière analogue ${}^\bullet p$ et $p^\bullet$. Lorsque chaque place a au plus une transition antécédent et une transition successeur, le réseau est un *graphe d'événements* et on peut noter ${}^\bullet p$ et $p^\bullet$ les uniques transitions précédant et suivant la place p si elles existent. La quantité $\mu(p)$ est appelé *marquage* (initial) de p. Elle dénote le nombre de *jetons* présents dans la place p. On représente graphiquement les places par des cercles contenant des jetons et les transitions par des rectangles.

Définition (Évolution d'un réseau de Petri)

Un réseau de Pétri *évolue* par déplacement des jetons entre les places selon les transitions. Plus précisément, une transition t est *franchissable* si $\forall p \in {}^\bullet t, \mu(p) \geqslant 1$ et on effectue une telle transition (on dit qu'on *tire* la transition) en retirant un jeton de toutes les places de ${}^\bullet t$ et ajoutant un jeton dans toutes les places de $t^\bullet$.

Exercice 2

Donner l'évolution des réseaux de Petri suivants. Quelles sont leurs différences ?



Exercice 3

Construire des réseaux de Petri qui effectuent les opérations suivantes. Dans la mesure du possible, essayer de se restreindre aux graphes d'événements.

1. choix non déterministe
2. addition du nombre de jetons situés dans deux places
3. soustraction d'une constante
4. multiplication par une constante
5. division par une constante
6. un compteur binaire
7. le schéma d'attente d'une file M/M/1

Exercice 4 (Accessibilité)

Un marquage est dit *accessible* s'il existe une évolution du réseau de Petri vers ce marquage.

1. Dans le cas général, le problème de l'accessibilité est EXPSPACE-difficile. Donner un semi-algorithme pour le résoudre.
2. Montrer que si l'on retire la contrainte de positivité du marquage, alors on peut résoudre le problème de l'accessibilité en temps polynomial avec des outils d'algèbre linéaire. En déduire une méthode de preuve d'inaccessibilité.

Exercice 5 (Réseaux de Petri bornés)

Un réseau de Petri est dit M-borné si le nombre de jetons dans chaque place ne peut dépasser M.

1. Comment caractériser les réseaux M-bornés en terme de marquages accessibles ?
2. En déduire un algorithme pour déterminer si un réseau de Petri est M-borné. Quel est sa complexité ?
3. Illustrer l'algorithme sur les exemples de l'exercice 2.
4. Déterminer une transformation entre réseaux de Petri qui force une place d'un réseau général à être M-bornée. Montrer la correction de la transformation.

Exercice 6 (Simulation des machines de Turing à mémoire bornée)

Vu les règles d'évolution des réseaux de Pétri, on se convainc sans peine qu'ils sont simulables par une machine de Turing. On va ici montrer une forme de réciproque : les réseaux de Petri peuvent simuler les machines de Turing à espace borné. La taille d'une machine de Turing va être la taille de son encodage plus la taille de l'espace mémoire qu'elle utilise. La simulation va produire un réseau de Petri 1-borné mais qui ne sera pas un graphe d'événement.

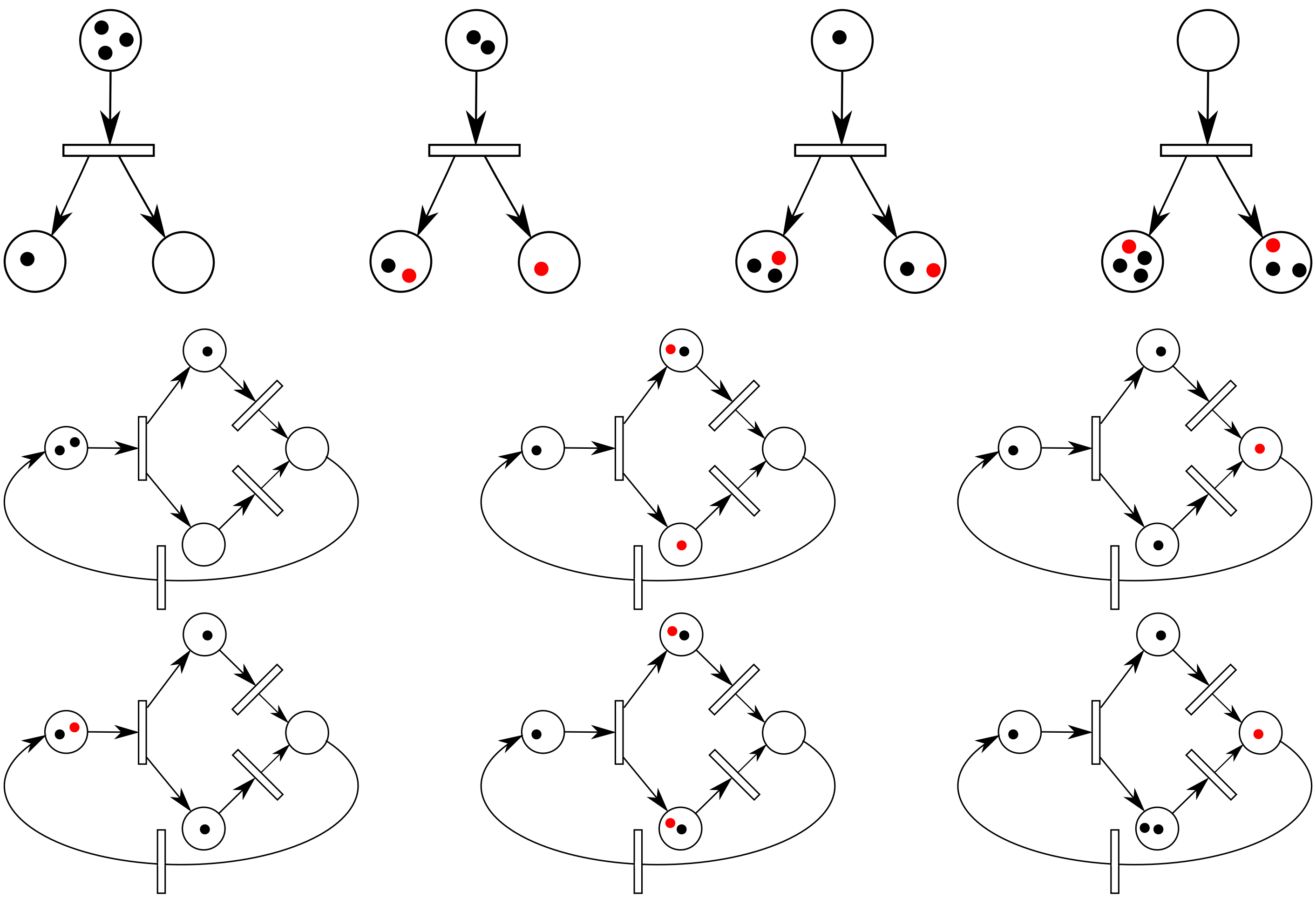
1. En utilisant un codage «1-hot» (donc presque unaire) des configurations de la machine de Turing, définir l'ensemble des places du réseau de simulation.
2. Donner à présent les transitions du réseau de Petri.
3. Préciser enfin son marquage initial.
4. Montrer que la taille du réseau de Petri obtenu est quadratique en la taille de la machine de Turing simulée.
5. En utilisant le fait que le problème de l'acceptation d'un mot par une machine de Turing à mémoire bornée est PSPACE-complet, comment en déduire la PSPACE-difficulté des problèmes suivants ?
   – accessibilité
   – existence d'une exécution infinie
   – évitement d'une place (aucun jeton n'y est jamais ajouté)
   – ...

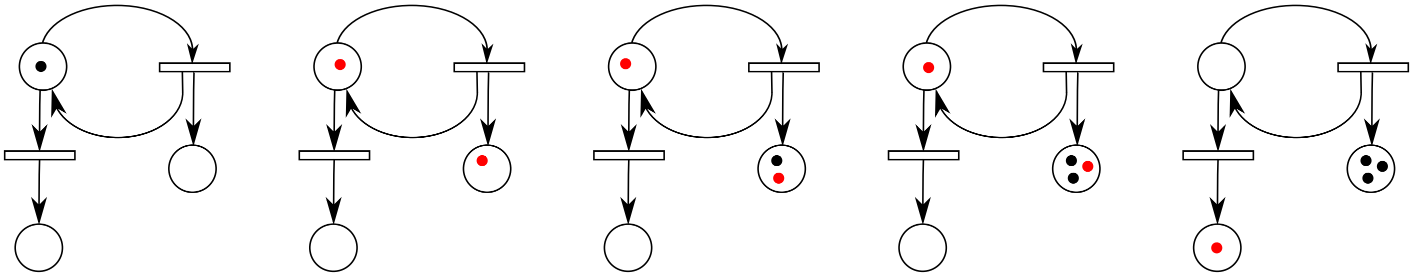
Solutions

▸ **Exercice 1**

1. horizontalement, $\mathcal{A} = \sum_{i=1}^{A(t)} W_i$
   verticalement, $\mathcal{A} = \int_0^t C(s)\,\mathrm{d}s$
2. $$W(t) = \frac{\sum_{i=1}^{A(t)} W_i}{A(t)} \qquad \lambda(t) = \frac{A(t)}{t} \qquad N(t) = \frac{1}{t}\int_0^t C(s)\,\mathrm{d}s$$
3. $$N(t) = \frac{\mathcal{A}}{t} = \frac{\sum_{i=1}^{A(t)} W_i}{t} = \frac{\sum_{i=1}^{A(t)} W_i}{A(t)}\frac{A(t)}{t} = \lambda(t)W(t)$$
   et en passant à la limite, on obtient $\overline{N} = \overline{\lambda}\,\overline{W}$.
4. Le théorème ergodique nous dit que la moyenne « spatiale » est égale à la moyenne temporelle. Dans le cas d'un régime permanent (pour que λ ait un sens), cela donne donc $\mathbb{E}(N) = \lambda\,\mathbb{E}(W)$. Le théorème de Little s'applique à toutes les chaînes ergodiques (pour lesquelles le théorème ergodique s'applique), *i.e.* les chaînes irréductibles et récurrentes t-positives.

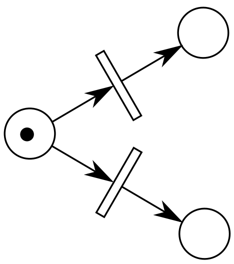
▸ **Exercice 2**

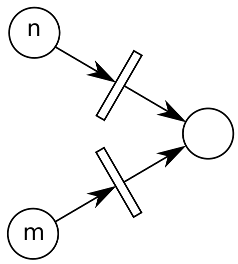


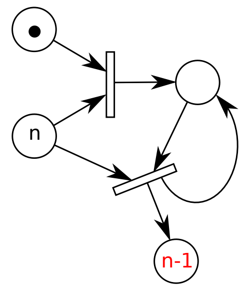


Leurs différences sont :
- la présence ou l'absence d'interbloquages (l'existence d'une exécution infinie)
- l'existence ou non de concurrence (plusieurs transitions qui s'excluent mutuellement)
- l'explosion du nombre de jetons

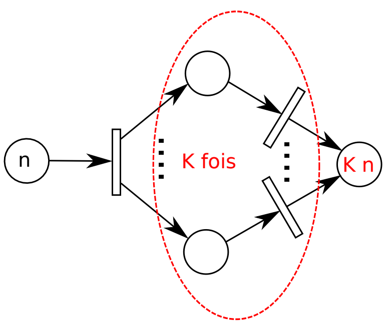
▸ **Exercice 3**

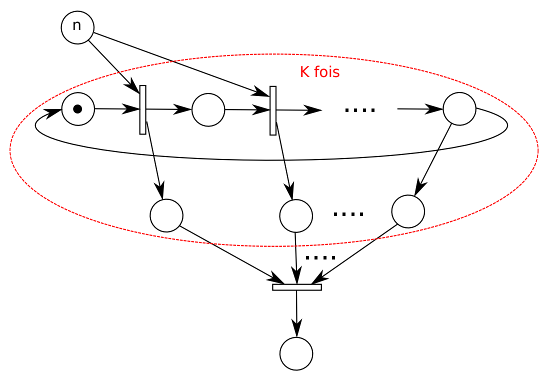
1. 

2. 

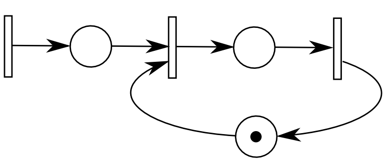
3. 

Il suffit de savoir faire la décrémentation (ci-dessus) puis d'itérer.

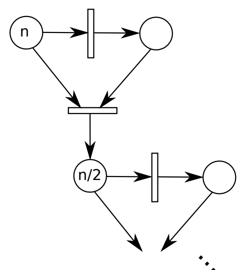
4. 

5. 

Si on renverse simplement les arcs de la multiplication, on obtient un réseau qui peut effectuer la division mais pas sur toutes ses exécutions. On utilise donc un jeton circulant pour n'autoriser qu'une unique transition à chaque étape.

7. 

On ne représente ici que deux contraintes :
- une unique personne par serveur
- les arrivées sont exogènes d'où l'utilisation d'une transition sans prédécesseur (qui peut donc être tirée à tout instant)

6. 

On utilise une succession de division par deux. Le jeton circulant devient alors un cas particulier de la méthode de l'exercice 5 pour borner une place.

► **Exercice 4**

1. On calcule le graphe des marquages accessibles (en largeur, pas en profondeur) et on regarde si le marquage recherché y apparaît.
2. Soit M la matrice de taille $|\mathcal{P}| \times |\mathcal{T}|$ dont le terme $m_{i,j}$ est $+1$ si $t_j \in {}^\bullet p_i$ et -1 si $t_j \in p_i^\bullet$ (et 0 sinon ou si $t_j \in {}^\bullet p_i \cap p_i^\bullet$). Dans ce cadre, tirer la transition t_j correspond à ajouter Me_j au vecteur $\overline{\mu} = (\mu(p_1), \ldots, \mu(p_{|\mathcal{P}|}))$. Déterminer si l'on peut passer de $\overline{\mu}_1$ à $\overline{\mu}_2$ revient à résoudre le système linéaire $MX = \overline{\mu}_2 - \overline{\mu}_1$, ce qui peut se faire en temps cubique.

   On peut remarquer que la matrice M ne caractérise pas le réseau de Petri, au contraire des matrices binaires de pré- et post-incidence M^- et M^+, de taille $|\mathcal{P}| \times |\mathcal{T}|$ et qui contiennent un 1 lorsque $t_j \in {}^\bullet p_i$ (resp. $t_j \in p_i^\bullet$). On a par ailleurs, $M = M^+ - M^-$.

► **Exercice 5**

1. Les réseaux de Petri bornés n'ont qu'un nombre fini de marquages accessibles. Ce nombre est même borné par $(M+1)^{|\mathcal{P}|}$.
2. Il suffit de calculer l'ensemble des configurations accessibles. Si cet ensemble est de cardinal trop grand ou qu'une coordonnée d'un marquage accessible dépasse la borne, alors il n'est pas M-borné. En fait, la seconde condition se produira toujours avant la première (ou au pire au même moment), de sorte qu'elle est suffisante pour définir l'algorithme.
3. Pour chaque place p, on ajoute une contre-place $\overline{p}$ telle que $t \in {}^\bullet\overline{p} \iff t \in p^\bullet$ et $t \in \overline{p}^\bullet \iff t \in {}^\bullet p$, *i.e.* on inverse le sens d'interaction des transitions voisines de p. Le marquage initial de $\overline{p}$ vaut $\mu_0(\overline{p}) = M - \mu_0(p)$. On a alors l'invariant $\mu(p + \mu(\overline{p}) = M$ et la condition de positivité d'un marquage assure la M-bornitude.

► **Exercice 6** Donnons-nous tout d'abord une machine de Turing $M = (Q, \Sigma, \Gamma, \Delta, q_0, F)$ utilisant une mémoire de taille n. Le codage «1-hot» revient à faire le produit de plusieurs codages unaires dans lesquels la configuration valide sera celle avec un jeton.

1. L'idée est de séparer les différents composants d'une configuration de M en différents types de places. Une configuration $C = (q, p, w)$ comporte l'état dans lequel on se trouve, la position de la tête de lecture et le mot écrit sur le ruban. L'ensemble des places est donc subdivisé en trois sous-parties :

   $E(q)$ $q \in Q$: les $|Q|$ places qui dénotent l'état de la machine de Turing

   $P(p)$ $1 \leqslant p \leqslant n$: la position de la tête de lecture

   $R(s, p)$ $s \in \Gamma, 1 \leqslant p \leqslant n$: la case p du ruban contient le symbole s
2. Les transitions vont synchroniser les différents éléments nécessaires à une transisiton de la machine de Turing M : pour chaque transition $(q, a, q', a', D) \in \Delta$, on crée n transitions t_p telles que :
   - ${}^\bullet t_p = \{E(q), P(p), R(a, p)\}$: on part de l'état q, en lisant la lettre a à la position p
   - $t_p^\bullet = \{E(q'), P(p+D), R(a', p)\}$: on arrive à l'état q', après avoir écrit la lettre a' en position p et on déplace la tête de lecture selon la valeur de D
3. Le marquage initial est nul sauf pour les places $P(q_0)$, $P(0)$ et $R(w_i, i)$ (avec w l'argument de M) où il vaut 1.
4. Il y a $|Q| + n + n \cdot |\Gamma|$ places et $n \cdot |\Delta|$ transitions ce qui donne $6n \cdot |\Delta|$ arcs. Au total, la simulation est quadratique en la taille de M, à savoir $O(n + |\Delta|)$ puisque $|Q|, |\Gamma| \leqslant |\Delta|$. Le réseau de Petri simule M en un sens très fort puisque qu'une transition dans la machine de Turing équivaut à une transition dans le réseau de Petri. On a donc bijection entre les exécutions des deux modèles.
5. Il faut simplement penser lors de la réduction à mentioner le fait que la construction du réseau de simulation soit polynomiale (en espace) en la taille de la machine initiale autorise à faire des réductions PSPACE.