

Réseaux de Pétri

TD n°1 et 2 - Hoerdet Mickaël/Quirin Arnaud - Mars 2004

1 Réseaux de Pétri

- ⇒ Un réseau de Pétri est composé de :
 - places
 - transitions
 - arêtes orientées
 - marquage initial
- ⇒ Permet de décrire l'évolution d'un système : les transitions ont des places d'entrée et des places de sortie.
- ⇒ Une transition est accessible ssi il y a au moins un jeton sur chacune de ses places d'entrée.
- ⇒ Si une transition est franchie, alors un jeton est pris à chacune des places d'entrée et un jeton est ajouté à chacune des places de sortie.

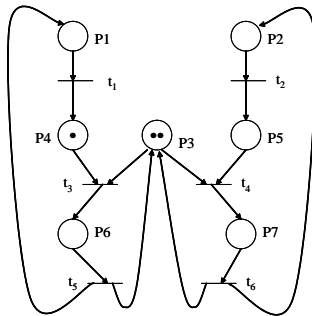


FIG. 1 – Exemple de réseau de Pétri

Marquage initial (P1, ..., P7) : (0,0,2,1,0,0,0)
 ⇒ t_3 est accessible
 ⇒ franchissement de t_3
 ⇒ nouveau marquage (0,0,1,0,0,1,0)

Rq :

Le nb de jetons du système n'est pas constant.

- ⇒ Cas de plus d'une transition accessible

Ex :

Marquage initial (0,0,1,1,0,0)

⇒ t_3 et t_4 sont accessibles

→ choix non déterministe

⇒ 2 marquages possibles après franchissement (0,0,0,0,1,1,0) ou (0,0,0,1,0,0,1)

- ⇒ Les réseaux de Pétri permettent de décrire des systèmes concurrents. Dans l'exemple :

- partie S1 : P1, P4, P6, t_1 , t_3 , t_5

- partie S2 : P2, P5, P7, t_2 , t_4 , t_6

- P3 : ressource partagée par S1 et S2

S1 et S2 fonctionnent de manière asynchrone (ordre de passage des transitions entre les deux systèmes quelconque). S1 et S2 se synchronisent au niveau de la ressource P3 (P3 accessible ⇒ il possède un jeton). Si P3 possède un seul jeton, alors S1 et S2 sont en conflit pour la ressource P3.

- ⇒ Propriétés caractérisant les réseaux de Pétri :

famine : dans l'exemple, le processus de gauche ou de droite ne s'exécute jamais : le système n'est pas équitable pour l'accès à la variable P3.

blocage : le réseau entre alors dans un état où plus aucune transition n'est possible. Si toutes les transitions restent accessibles, le réseau est dit vivace.

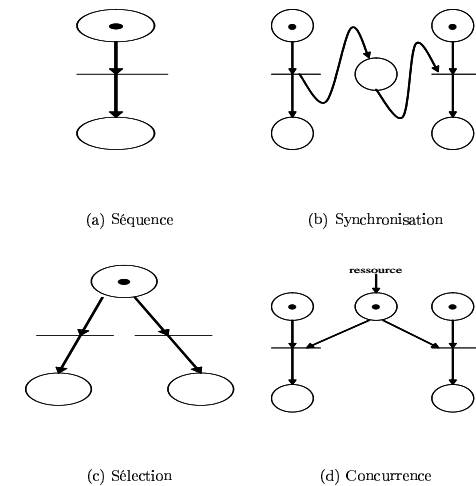


FIG. 2 – Exemples de réseaux de Pétri simples

1.1 Questions introductives

Expliciter $M(P3)$, $M0$, $*M0$, ${}^{\circ}t_3$, t_3° , ${}^{\circ}P3$, $P3^{\circ}$ sur la Fig. 1.

2 Exercices

2.1 Programmes communicants

Soit 2 programmes exécutés sur 2 processeurs différents P1 et P2 d'un ordinateur parallèle :

Processeur P1: debut : si test1 = vrai alors envoie(m,P2) sinon reception(m,P2) finsi aller à debut	Processeur P2: debut: si test2 = vrai alors envoie(m,P1) sinon reception(m,P1) finsi aller à début
--------------------------------------------------------------------------------------------------------------------	-------------------------------------------------------------------------------------------------------------------

L'envoi d'un message m ne peut se faire que lorsque le destinataire est en attente de réception d'un message.

2.1.1 Questions

1. Représenter le comportement de ces deux programmes par un réseau de Pétri.
2. Peut-il avoir blocage? Si oui, dans quel cas?
3. Proposer une simplification.
4. Le réseau est-il vivace?
5. Faites le graphe de marquage du réseau simplifié.

2.2 Programmation d'un distributeur de boissons

Dans son état initial il attend qu'on appuie sur le bouton "**café**" ou "**sirop**". L'utilisateur peut changer d'avis s'il s'est trompé. Après cette sélection (et changement d'avis potentiel), il peut "**confirmer**" ou "**annuler**". Si on confirme le distributeur a besoin de quelques secondes pour verser la boisson pendant lesquelles tous les boutons sont inactifs. Quand la boisson est prête ou si on a annulé, il arrive dans son état final qui est en même temps l'état de départ.

2.2.1 Questions

1. Modéliser le comportement du distributeur par un réseau de Pétri.
2. Ce réseau est-il autonome?
3. Rajouter la gestion de la disponibilité d'une boisson. Si une boisson n'est plus disponible, la machine l'indique et elle ne la distribue plus jusqu'au prochain approvisionnement.
4. Rajouter la gestion du paiement. On ne donne une boisson que si l'utilisateur a introduit une somme d'argent suffisante. L'utilisateur peut récupérer son argent sans prendre de boisson. Lorsque la caisse est pleine la machine ne peut fonctionner.

2.3 Mécanisme d'alimentation papier d'une imprimante

Cette imprimante est destinée à être connectée localement à un micro-ordinateur. Le papier est stockée dans un chargeur sur le dessus de l'imprimante. Le mécanisme d'alimentation papier essaye de charger du papier dans l'imprimante lorsque l'utilisateur appuie sur le bouton "**load**", lorsque l'imprimante reçoit une donnée quelconque de l'ordinateur ou lorsque le processus d'impression a atteint la fin de la page courante. Si le papier a été chargé avec succès, l'imprimante est prête à débiter l'impression de la page suivante. Si le papier n'a pas été chargé correctement : soit le chargeur papier est vide, soit l'imprimante est considérée en état "**bourrage papier**".

- **bourrage** : Un voyant lumineux bourrage clignote, aucune donnée n'est traitée avant que l'utilisateur ne résolve le problème et appuie sur le bouton "**load**".
- **chargeur vide** : le voyant papier clignote, aucune donnée n'est traitée avant que l'utilisateur ne remette du papier dans le chargeur.

Si des données sont reçues en cas de chargeur vide ou de de bourrage, un voyant d'erreur supplémentaire se met à clignoter. Lorsque le papier a été chargé et que l'impression a démarré, le mécanisme d'alimentation papier attend une commande de fin de ligne de la part de l'ordinateur. A la réception de cette commande le mécanisme avance le papier d'une ligne à moins qu'il ne soit déjà positionné en fin de feuille de papier auquel cas il tente de charger la feuille suivante.

2.3.1 Question

1. Modéliser le comportement du distributeur par un réseau de Pétri.