

# Thème 11

## Réseaux de Petri Places-Transitions

---

### Contenu du thème

1. Introduction
2. RdP PT
3. Protocoles de communication

### Références

Diaz, Michel (2001) **Les Réseaux de Petri – Modèles fondamentaux**, Hermes Science Publications : Paris.

Peterson, J. L. (1977) « Petri Nets », **Computing Surveys**, vol. 9, N° 3, pp 223-252.

<http://www.daimi.au.dk/PetriNets/>

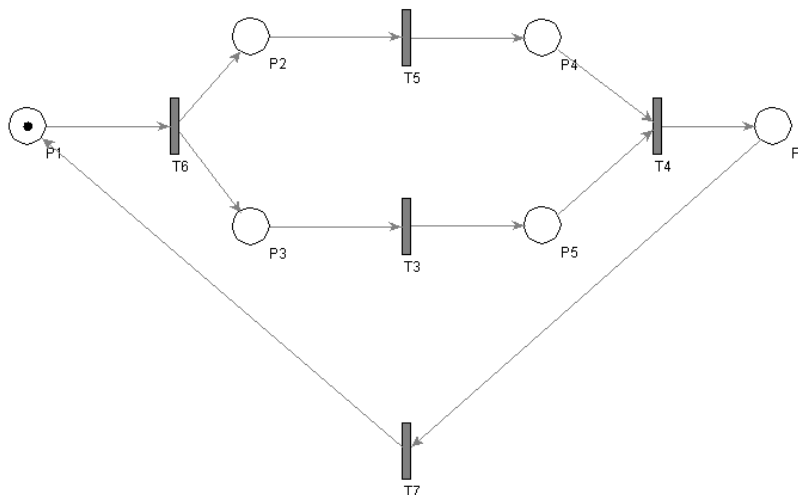
## 1 Introduction

Créés par Carl Adam Petri en 1962 afin de modéliser la **composition** et la **communication** entre automates.

Le 1<sup>er</sup> modèle, appelé RdP CE (conditions-événements) reposait sur l'utilisation des valeurs booléennes, vrai et faux.

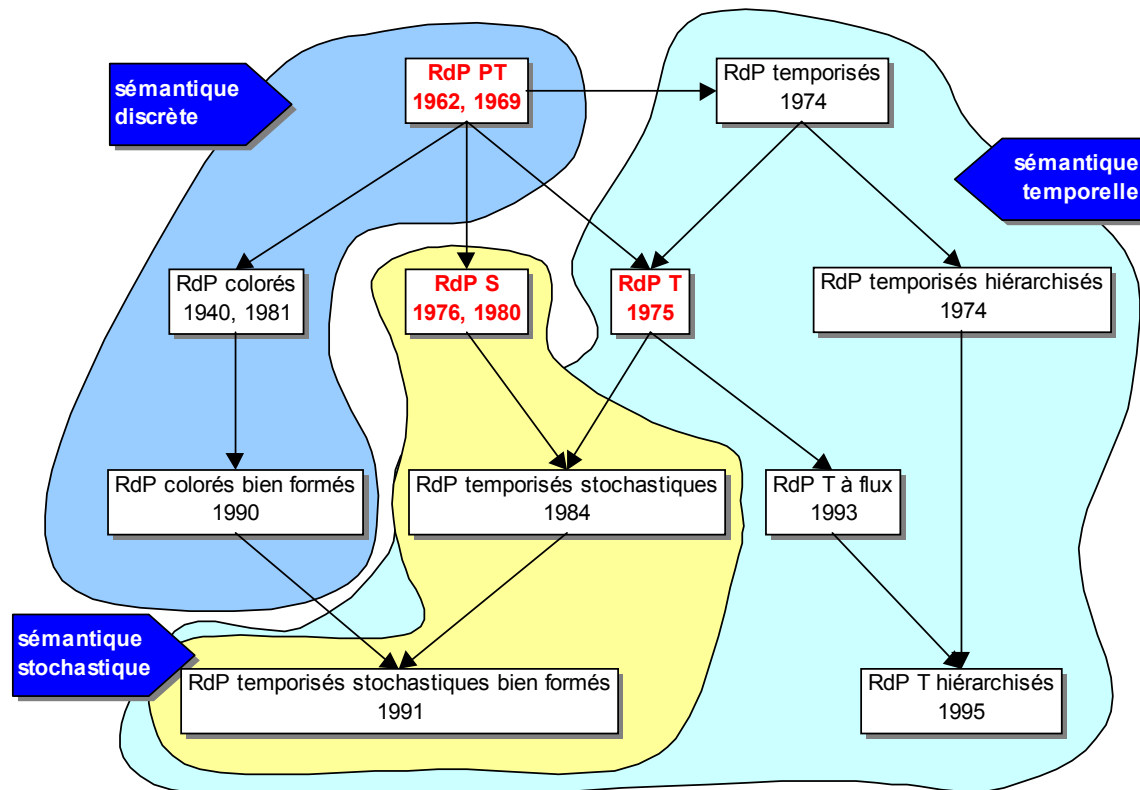
Les RdP PT (places-transitions) généralisent les RdP CE en manipulant des valeurs entières.

### Exemple 1 (sauvegardé en format GIF avec JARP)



**Éléments de vocabulaire** : les jetons, le marquage, les places, les transitions, le tir ou le franchissement d'une transition, transition tirable ou franchissable.

## Les familles de RdP :



Trois grandes familles (qui ne sont évidemment pas étanches) peuvent se distinguer. Elles correspondent à trois domaines sémantiques : la sémantique discrète (RdP PT et ses dérivés) pour les comportements qui peuvent se représenter par des graphes finis ou dénombrables d'états; la sémantique continue (RdP T et ses dérivés) pour les comportements qui nécessitent la prise en considération explicite d'un temps dense; la sémantique stochastique (RdP S et ses dérivés) pour les comportements qui incluent des distributions de franchissement et conduisent à des processus stochastiques (chaînes de Markov, etc.).

Quelques **outils de base** pour manipuler des RdP (disponibles sur le site « Petri Nets World »):

- ◇ JARP : éditeur graphique, simulation, sauvegarde en GIF, JPEG, etc. Plate-forme Java.
- ◇ RENEW : éditeur graphique, simulation, sauvegarde EPS, PS, XML. Plate-forme Java.
- ◇ PetriNet Kernel (PNK) : éditeur graphique (très peu intuitive), simulation. Plate-forme Java.
- ◇ INA : PAS d'éditeur graphique, analyses poussées. Plate-forme Windows.
- ◇ SIMPRES : éditeur graphique, simulation. Plate-forme Java.
- ◇ Danamics : éditeur graphique, simulation, analyses simples. Plate-forme Java.

## 2 RdP PT

### 2.1 Automates et machines à états

**Rappel :** Les modèles fondés sur les automates et les machines à états reposent sur trois hypothèses.

#### Hypothèse 1

Il existe, dans le système modélisé, une notion d'état global, un ensemble de ces états et une représentation explicite de ces états.

#### Hypothèse 2

Le fonctionnement (ou le comportement) du système modélisé peut se décrire par des règles permettant de passer d'un état global de départ (donné) à un autre état global d'arrivée (donnée). La transition s'effectue quand survient un événement conditionnant l'évolution du système.

À partir de ces deux hypothèses, la description du système est définie par un graphe global qui représente le fonctionnement du système.

#### Hypothèse 3

C'est l'hypothèse de l'indivisibilité des transitions entre états. Lorsqu'un événement survient, celui-ci doit être totalement terminé avant l'arrivée d'un autre événement.

Implication de cette hypothèse sur la signification des modèles : Quand des actions complexes sont associées aux transitions, elles doivent être exécutées complètement avant d'atteindre l'état suivant. En conséquence, il n'existe que deux états globaux, l'un avant, l'autre après. On ne peut pas dire : je franchis partiellement la transition X.

Conséquence sur le travail de modélisation : Les transitions modélisées doivent aussi être effectivement indivisibles dans le système considéré. Si certaines instructions peuvent être interrompues, la modélisation doit en tenir compte.

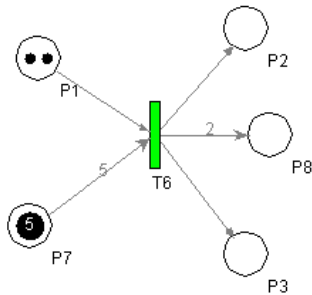
Ces trois hypothèses tiennent encore dans la modélisation par RdP.

### 2.2 Des machines à états aux RdP

Un réseau de petri est un modèle défini par :

- ◇ un ensemble de places, notées graphiquement par des cercles;
- ◇ un ensemble de transitions, notées graphiquement par des barres ou des rectangles;
- ◇ un ensemble d'arcs, notés par des flèches qui joignent les places aux transitions et les transitions aux places;
- ◇ une distribution de jetons dans les places.

## Exemple 2



Conséquence de la notion de transition : La transition permet ainsi de composer des machines à états.

Conséquence de la notion de marquage : Le système ne sera plus modélisé par la considération d'un état global mais par celle d'un ensemble de sous-états (places) spécifiques (liés aux jetons). L'ensemble des places marquées constituera l'état du RdP complet.

Conséquence de la notion de jeton : Une place peut représenter la consommation de « n » ressources ou, à l'inverse, la production de « n » ressources.

## 2.3 Concepts et définitions

### Définition 1

Un réseau de Petri Places-transitions R se définit comme un uplet :  $(P, T, \text{Pré}, \text{Post})$  où :

1. P est l'ensemble des places;
2. T est l'ensemble des transitions;
3.  $\text{Pré} = P \times T \mapsto \mathbb{N}$ , une application d'incidence avant où  $\text{Pré}(p, t)$  contient la valeur entière « n » associée à l'arc allant de « p » à « t ».
4.  $\text{Post} = P \times T \mapsto \mathbb{N}$ , une application d'incidence arrière où  $\text{Post}(p, t)$  contient la valeur entière « n » associée à l'arc allant de « t » à « p ».

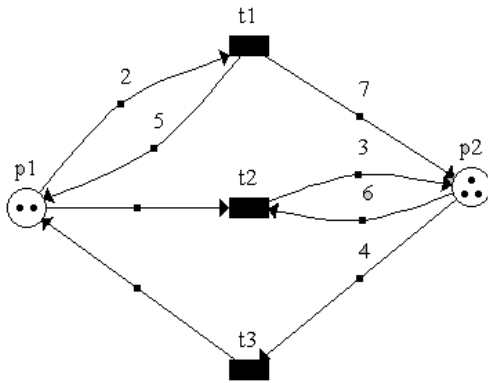
### À propos des places :

- ◇ « p » est une place d'entrée de la transition « t » si  $\text{Pré}(p, t) > 0$
- ◇ « p » est une place de sortie de la transition « t » si  $\text{Post}(p, t) > 0$

### Représentation matricielle

Il est facile de construire une représentation matricielle d'un RdP PT. On construit la matrice correspondant à Pré et celle correspondant à Post. Les lignes identifient les places et les colonnes, les transitions. Une intersection  $(i, j)$ , dans Pré, correspond au coût associé à l'arc menant de la place « i » à la transition « j », tandis qu'une intersection  $(i, j)$ , dans Post, correspond au coût associé à l'arc menant de la transition « j » à la place « i ».

### Exemple 3



		Pré			Post		
		t1	t2	t3	t1	t2	t3
cas 1	p1	2	1	0	5	0	1
	p2	0	6	4	7	3	0
		cas 3			cas 4		
		t1	t2	t3	t1	t2	t3
		cas 2	cas 1	cas 2	cas 1	cas 2	cas 1

Ce cette manière, il est facile d’aller chercher, tout ce qui sort d’une place (cas 1), tout ce qui entre dans une place (cas 2), tout ce qui entre dans une transition (cas 3) et tout ce qui sort d’une transition (cas 4).

### Définition 2

Le marquage d’un RdP est une application  $M : P \mapsto \mathbb{N}$  donnant pour chaque place le nombre de jetons qu’elle contient. Le marquage initial sera noté  $m_0$  et  $M(p)$  indique le marquage de la place « p ».

### Définition 3

Le fonctionnement d’un RdP se définit comme suit. Pour un marquage « M », une transition « t » est dite tirable (ou franchissable ou sensibilisée) SSI

$$\forall p_i \in P, \text{ on a } M(p_i) \geq \text{Pré}(p_i, t)$$

c’est-à-dire, pour toutes les places « p<sub>i</sub> », entrées de « t », le nombre de jetons dans « p<sub>i</sub> »,  $M(p_i)$ , est supérieur ou égal au poids de l’arc allant de « p<sub>i</sub> » à « t ».

Quelques notations supplémentaires :

- ◇ Le fait que « t » est tirable depuis « M » se notera :  $M[t\rangle$
- ◇ Le tir de « t » depuis « M » donnant le nouveau marquage « M' » se notera :  $M[t\rangle M'$

### Définition 4

Dans un RdP PT, toute transition tirable « t » peut être tirée et son tir conduit à un nouveau marquage  $M'$  définit par :

$$\forall p \in P, M'(p) = M(p) - \text{Pré}(p, t) + \text{Post}(p, t)$$

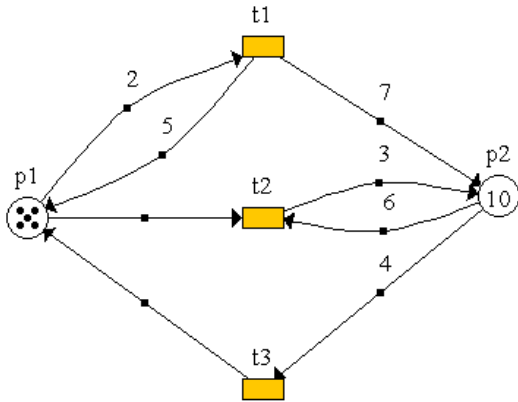
### Exemple d'un tir grâce à un calcul matriciel sur le réseau de l'exemple 3

On veut tirer t1, le nouveau marquage sera :

$$= \begin{bmatrix} 2 \\ 3 \end{bmatrix} - \begin{bmatrix} 2 & 1 & 0 \\ 0 & 6 & 4 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 5 & 0 & 1 \\ 7 & 3 & 0 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$$

$$= \begin{bmatrix} 2 \\ 3 \end{bmatrix} - \begin{bmatrix} 2 \\ 0 \end{bmatrix} + \begin{bmatrix} 5 \\ 7 \end{bmatrix} = \begin{bmatrix} 5 \\ 10 \end{bmatrix}$$

Ce qui donne graphiquement :



La définition 4 permet d'exprimer la synchronisation de base dans les systèmes :

- ◇ la causalité (un événement « t » précède toujours un autre événement « t' ») et le parallélisme;
- ◇ l'attente (par le manque de jetons);
- ◇ l'accroissement et la diminution du parallélisme et/ou des ressources;
- ◇ le non-déterminisme;
- ◇ les conflits.

On remarquera facilement que la section « Post – Pre » de l'équation de la définition 4 peut se simplifier en considérant la matrice d'incidence, C, du réseau construite en calculant d'abord la matrice correspondant à la section encadrée :

$$M' = M + \boxed{\text{Post}(p,t) - \text{Pré}(p, t)}$$

Nous avons d'ailleurs illustré comment, à partir d'un calcul matriciel, on peut facilement déduire le nouveau marquage « M' ».

#### Définition 5

La matrice d'incidence d'un réseau, notée C, est définie par :

$$\forall p \in P, \forall t \in T, C(p,t) = \text{Post}(p,t) - \text{Pré}(p,t)$$

Grâce à cette définition, on peut exprimer le calcul d'un nouveau marquage « M' », à partir d'un marquage « M », s'obtient, suite au tir de la transition « t », en calculant, pour toute place « p » du réseau :

$$M' = M + C(p,t)$$

### Suite de notre exemple

L'exemple du tir de la transition « t » peut donc s'exprimer avec la matrice d'incidence C :

$$= \begin{bmatrix} 2 \\ 3 \end{bmatrix} + \begin{bmatrix} 3 & -1 & 1 \\ 7 & -3 & -4 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 2 \\ 3 \end{bmatrix} + \begin{bmatrix} 3 \\ 7 \end{bmatrix} = \begin{bmatrix} 5 \\ 10 \end{bmatrix}$$

### Définition 6 : Équation fondamentale du franchissement

Ainsi, étant donnée une situation où :

$$M[t_1]M_1, M_1[t_2]M_2, \dots, M_n[t_n]M'$$

alors on obtient l'équation fondamentale du franchissement :

$$M' = M + C \cdot \overline{s}$$

où  $\overline{s}$  est le vecteur caractéristique de la séquence de transitions

$$s = t_1 t_2 \dots t_n$$

tel que  $\overline{s}(t)$  donne le nombre d'occurrences de la transition « t » dans « s ». On note :

$$M[s]M'$$

Ainsi, on peut facilement calculer le nouveau marquage induit par le franchissement de la séquence  $t_1 t_2 t_3$  dans le réseau précédent. On obtiendrait :

$$= \begin{bmatrix} 2 \\ 3 \end{bmatrix} + \begin{bmatrix} 3 & -1 & 1 \\ 7 & -3 & -4 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 2 \\ 3 \end{bmatrix} + \begin{bmatrix} 3 \\ 0 \end{bmatrix} = \begin{bmatrix} 5 \\ 3 \end{bmatrix}$$

Ou encore, si vous avez besoin d'un autre exemple, le nouveau marquage induit par le franchissement de la séquence  $t_1 t_2 t_3 t_1$  :

$$= \begin{bmatrix} 2 \\ 3 \end{bmatrix} + \begin{bmatrix} 3 & -1 & 1 \\ 7 & -3 & -4 \end{bmatrix} \begin{bmatrix} 2 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 2 \\ 3 \end{bmatrix} + \begin{bmatrix} 6 \\ 7 \end{bmatrix} = \begin{bmatrix} 8 \\ 10 \end{bmatrix}$$

## 2.4 Graphe d'accessibilité ou graphe de marquage

### Définition 7

L'ensemble des marquages accessibles  $A(R, m_0)$  se définit par le plus petit ensemble :

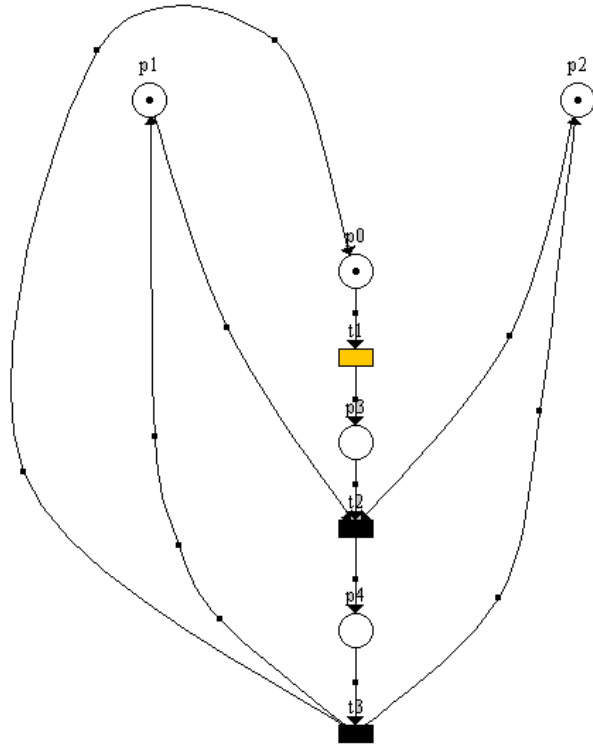
$$m_0 \in A \text{ et si } M \in A \text{ et } M \xrightarrow{-t} M' \text{ alors } M' \in A.$$

L'ensemble A définit l'ensemble des marquages accessibles par un ensemble de vecteurs qui indiquent le nombre de jetons dans chaque place. Exemple : le marquage actuel de l'exemple 3 après le tir de « t1 » est le vecteur : [5, 10].

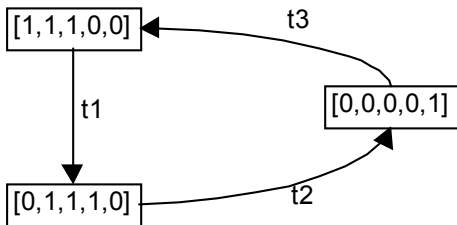
**Définition 8**

Le graphe des marquages accessibles  $G(R, m_0)$  est défini comme le graphe dont les nœuds sont les marquages accessibles de  $A(R, m_0)$  et dont les arcs sont les noms des transitions impliquées dans les tirs menant d'un marquage à un autre.

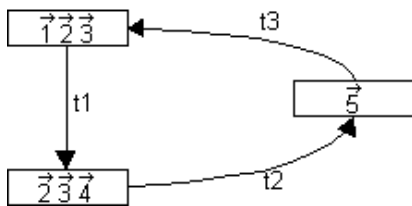
**Exemple 4**



Le graphe d'accessibilité de ce réseau est :



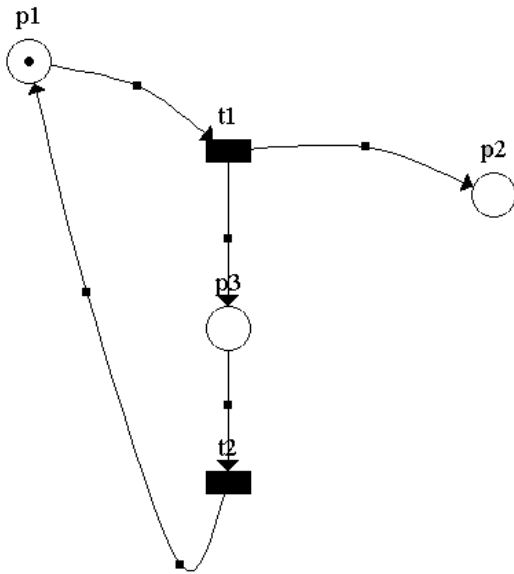
On peut également utiliser une autre notation. Notons  $\vec{i}$  pour indiquer que la place « i » est marquée par un jeton et  $n\vec{i}$ , si la place « i » est marquée de « n » jetons. Alors le graphe précédent peut s'exprimer comme suit :



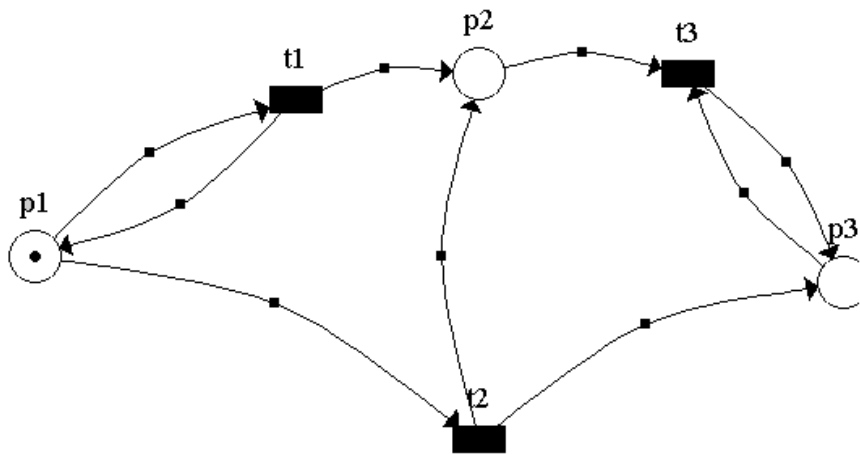


### Exemple 5

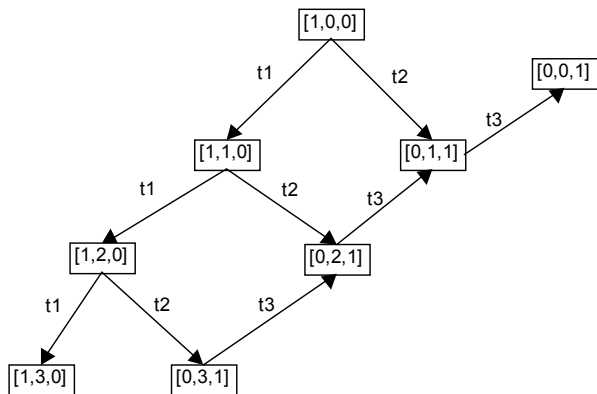
Évidemment, un graphe de marquage peut être infini. Il est facile de voir ce fait en examinant la place « p2 » qui ne fait que recevoir des jetons.



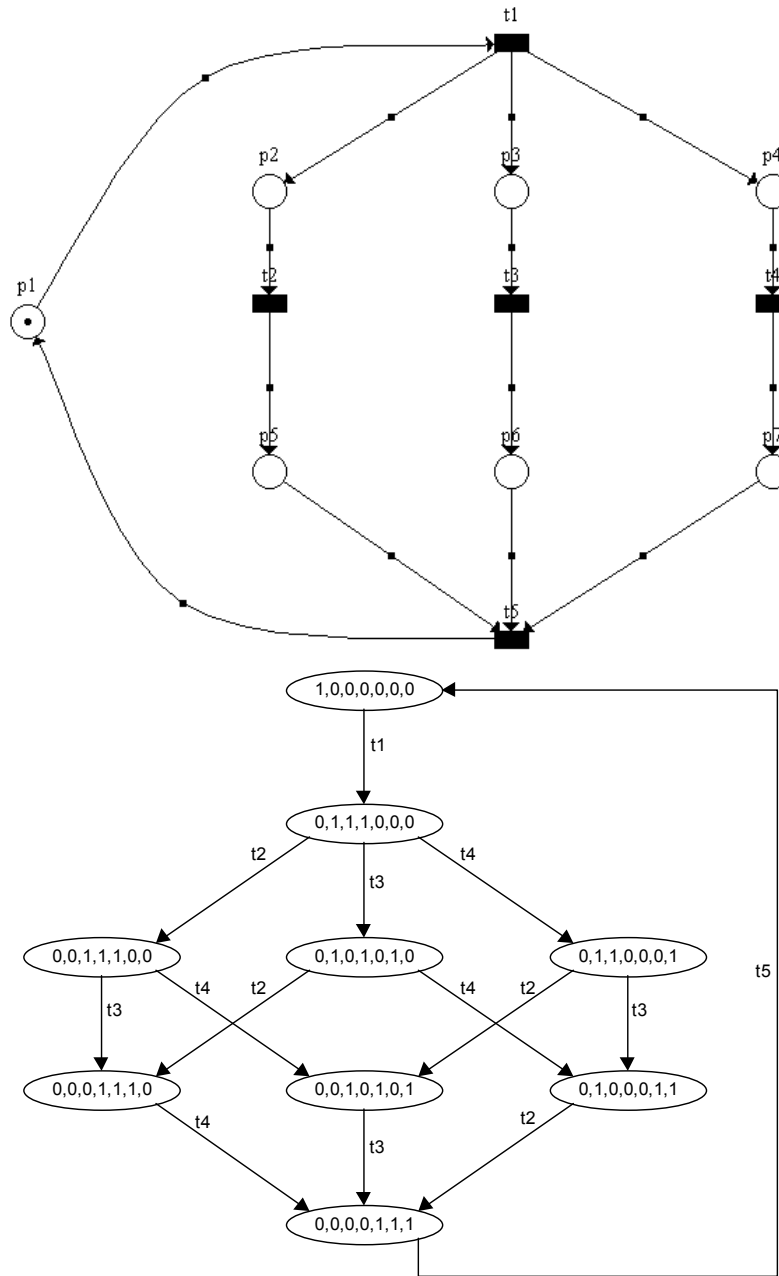
### Exemple 6



Le graphe d'accessibilité est encore une fois infini. En voici une partie :

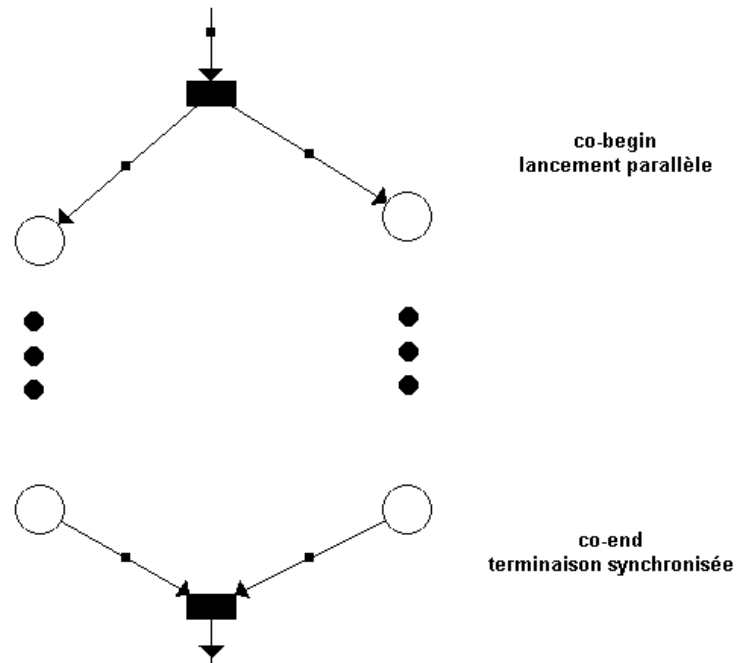


### Exemple 7



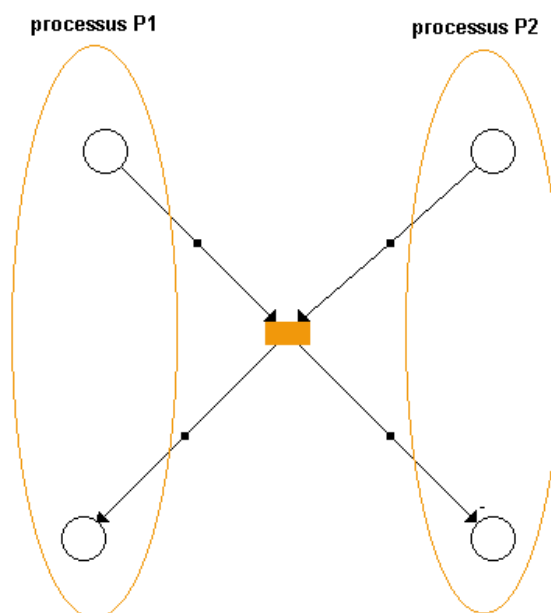
## 2.5 Schémas de synchronisation élémentaires

### 2.5.1 Parallélisme ou concurrence



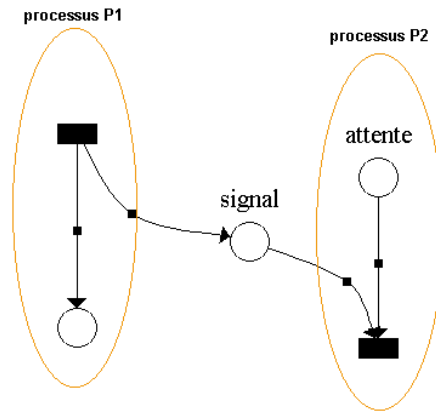
### 2.5.2 Synchronisation

#### A Par rendez-vous



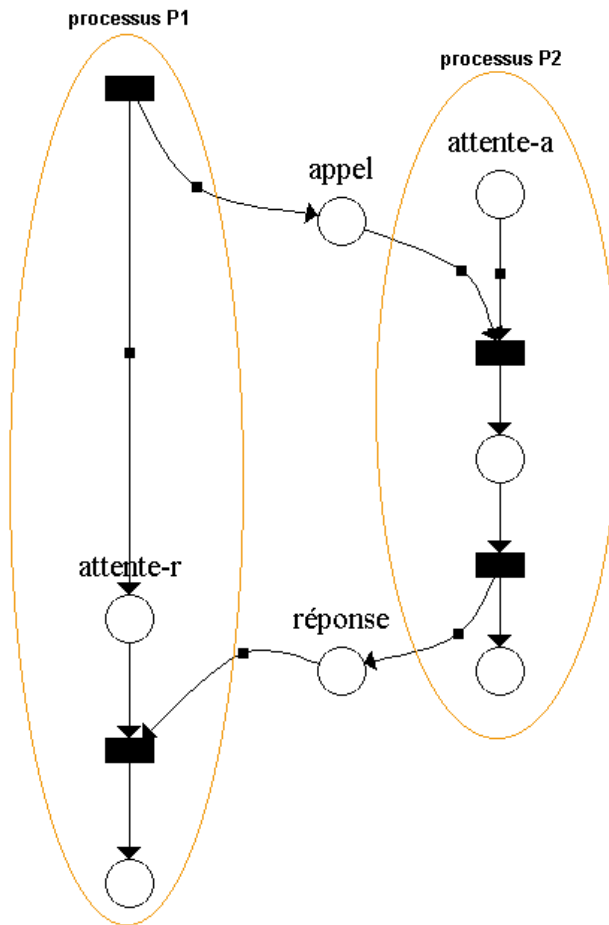
## B Par signal (sémaphore)

Si la place « signal » est marquée et la place « attente » ne l'est pas, cela signifie que le processus P1 a envoyé le signal mais le processus P2 ne l'a pas encore reçu. Si, par contre, la place « signal » n'est pas marquée et que la place « attente » est marquée, cela signifie que le processus P2 est en attente du signal.



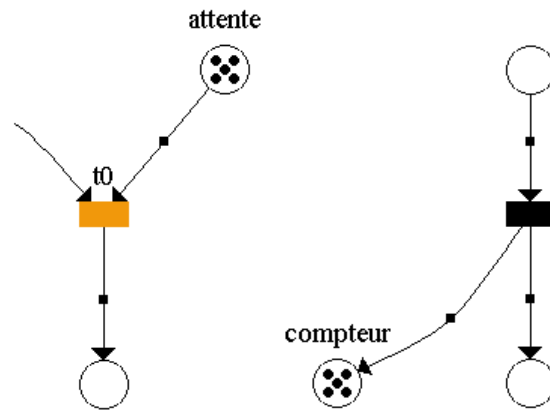
## C Par appel/réponse

Une variante du modèle précédent qui consiste à faire un signal d'appel provenant du processus P1 et un signal de réponse provenant du processus P2.



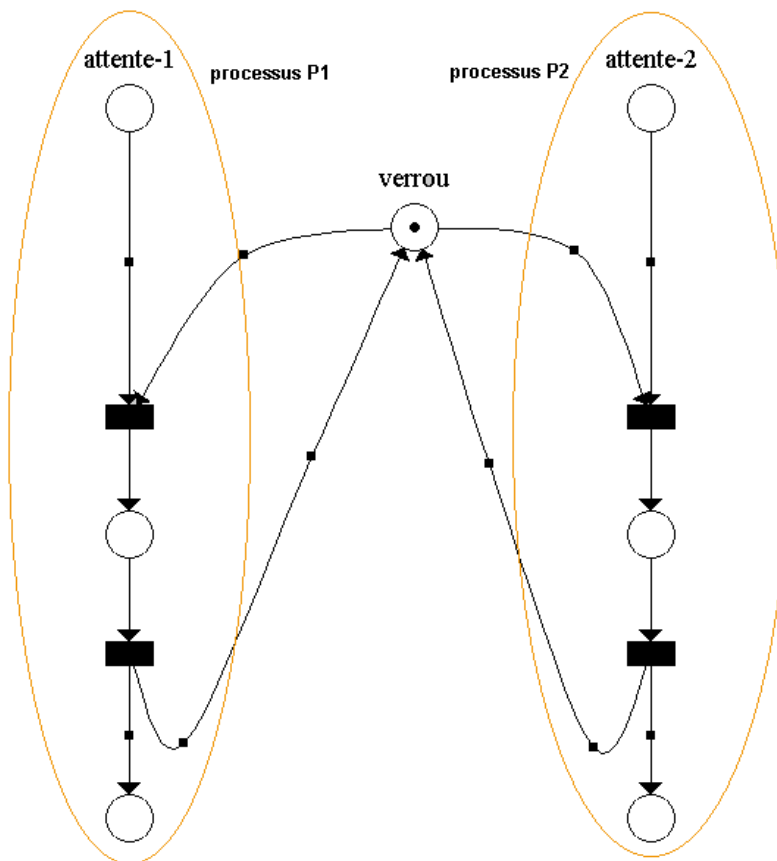
## D Mémorisation

Dans tous les modèles, on peut compter le nombre de tirs d'une transition en utilisant une place sans sortie. Évidemment, dans le cas des modèles B et C, les places « attente » peuvent servir à indiquer combien d'instances d'un processus sont en attente, exemplifiant ainsi la notion de « bottle neck ».



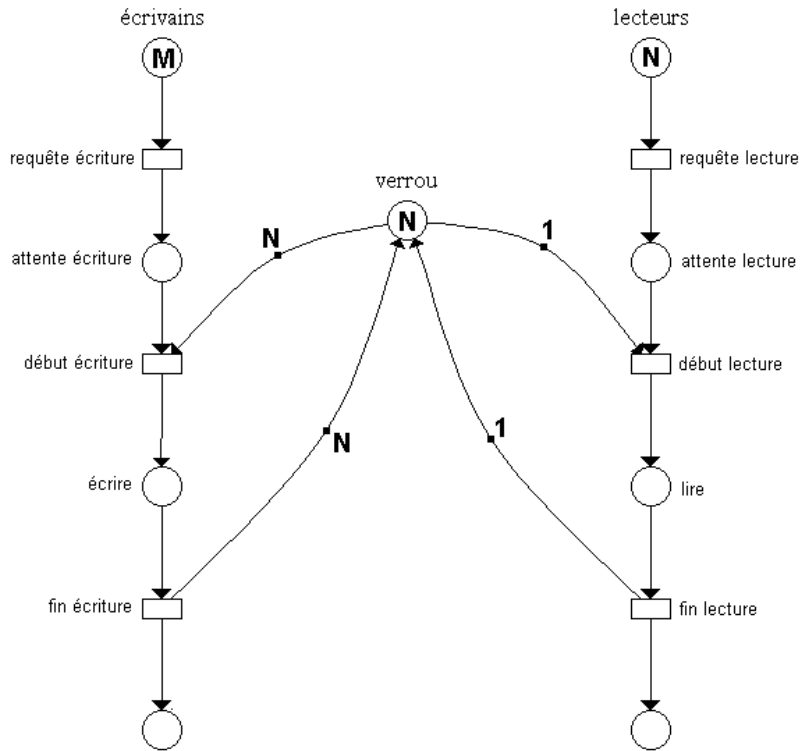
### E Exclusion mutuelle

La section critique de l'un ou l'autre des deux processus est exécutée de manière exclusive grâce à la place « verrou ».



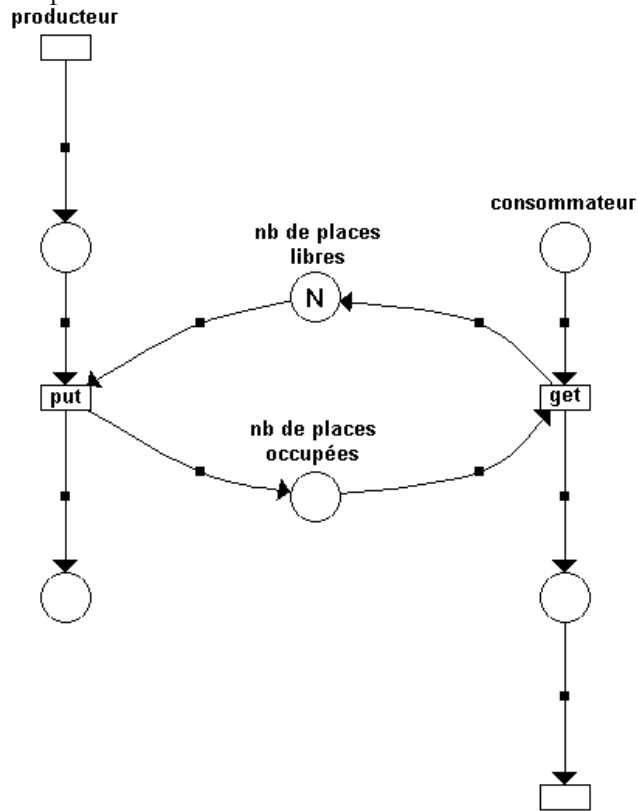
### F Lecteurs/écrivains

Il s'agit d'une variante du « verrou ». Ici on désire M écrivains et N lecteurs. Les écrivains ne peuvent écrire en même temps (donc sont en exclusion mutuelle). Les écrivains et les lecteurs ne peuvent écrire et lire en même temps (donc sont en exclusion mutuelle). Cependant, les lecteurs peuvent lire en parallèle. Voici le modèle simple de ce système.



### G Tampon borné de N espaces

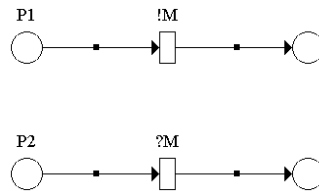
Une variante du modèle précédent.



### 3 Protocoles de communication

#### 3.1 Modèles de base

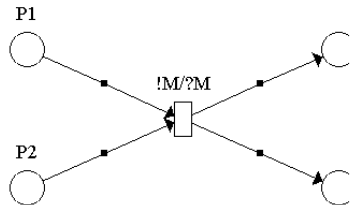
Prenons deux processus, P1 et P2 qui s'échange un message M. La notation !M indique l'envoi du message et ?M, la réception du message.



Il existe, d'après nos modèles précédents, trois manières de modéliser le système : par rendez-vous (fusion), par envoi de signal (place partagée) et par appel-réponse.

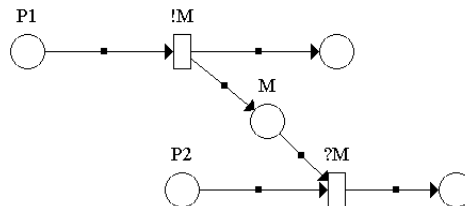
#### Solution A : fusion ou rendez-vous (solution symétrique)

Notons que lors de la synchronisation des deux processus, un passage de valeurs peut avoir lieu mais n'est pas représenté par ce modèle.



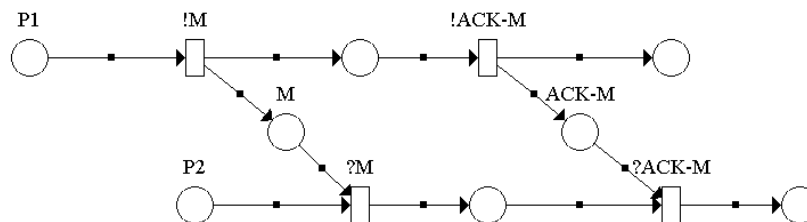
#### Solution B : envoi de signal ou place partagée (solution asymétrique)

Le marquage de la place M indique que le message a été envoyé mais non encore reçu.



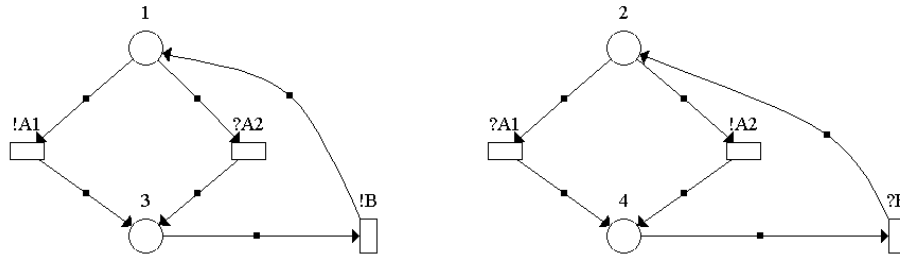
#### Solution C : appel-réponse (solution asymétrique)

Ce modèle ajoute et explicite le fait que l'émetteur attend une réponse avant de poursuivre son activité.

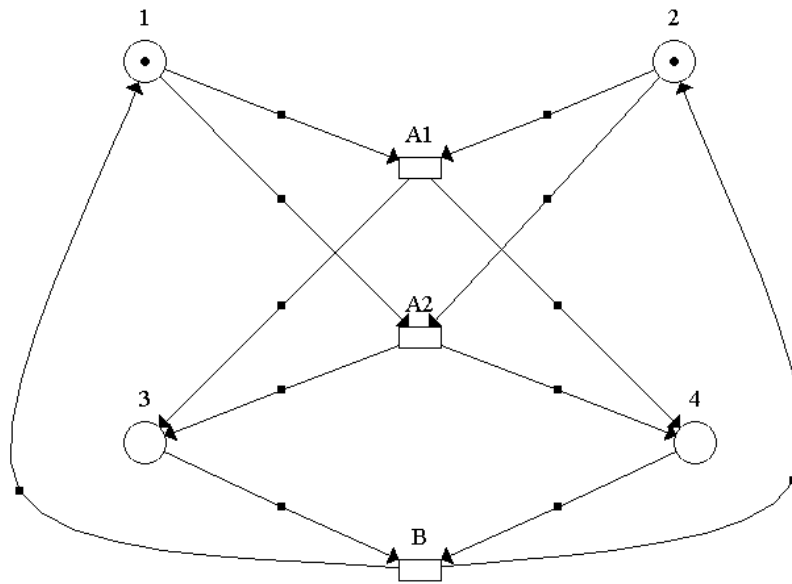


### 3.2 Ouverture simple d'une connexion

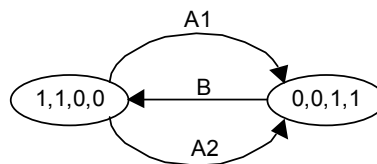
Le processus P1 peut ouvrir une connexion par envoi du message A1 et le processus P2 peut également ouvrir une connexion par envoi du message A2. Mais seul le processus P1 possède le droit de fermer la connexion ouverte par envoi du message B. Voici les deux processus :



**Solution A : rendez-vous**

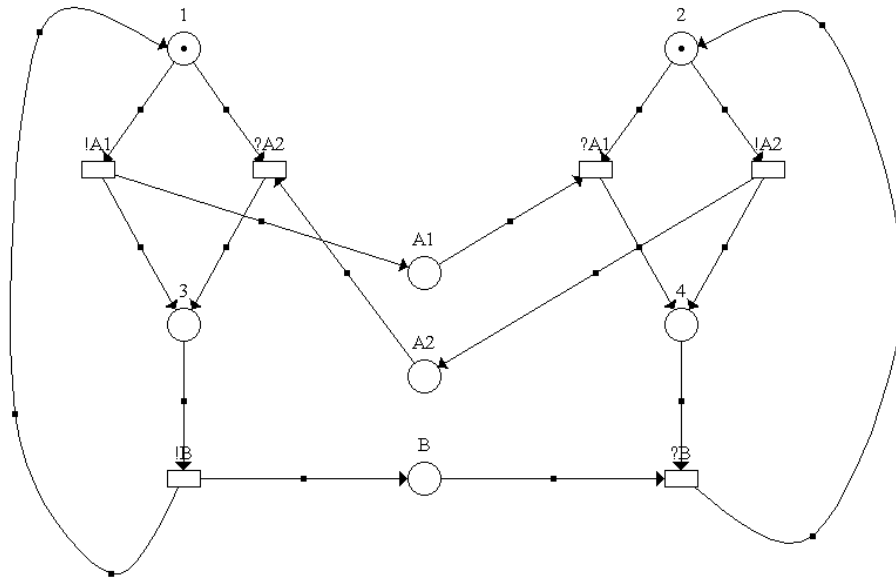


Le graphe d'accessibilité est normal.

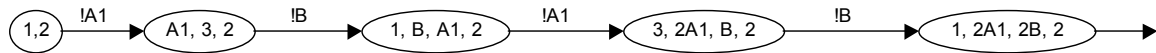




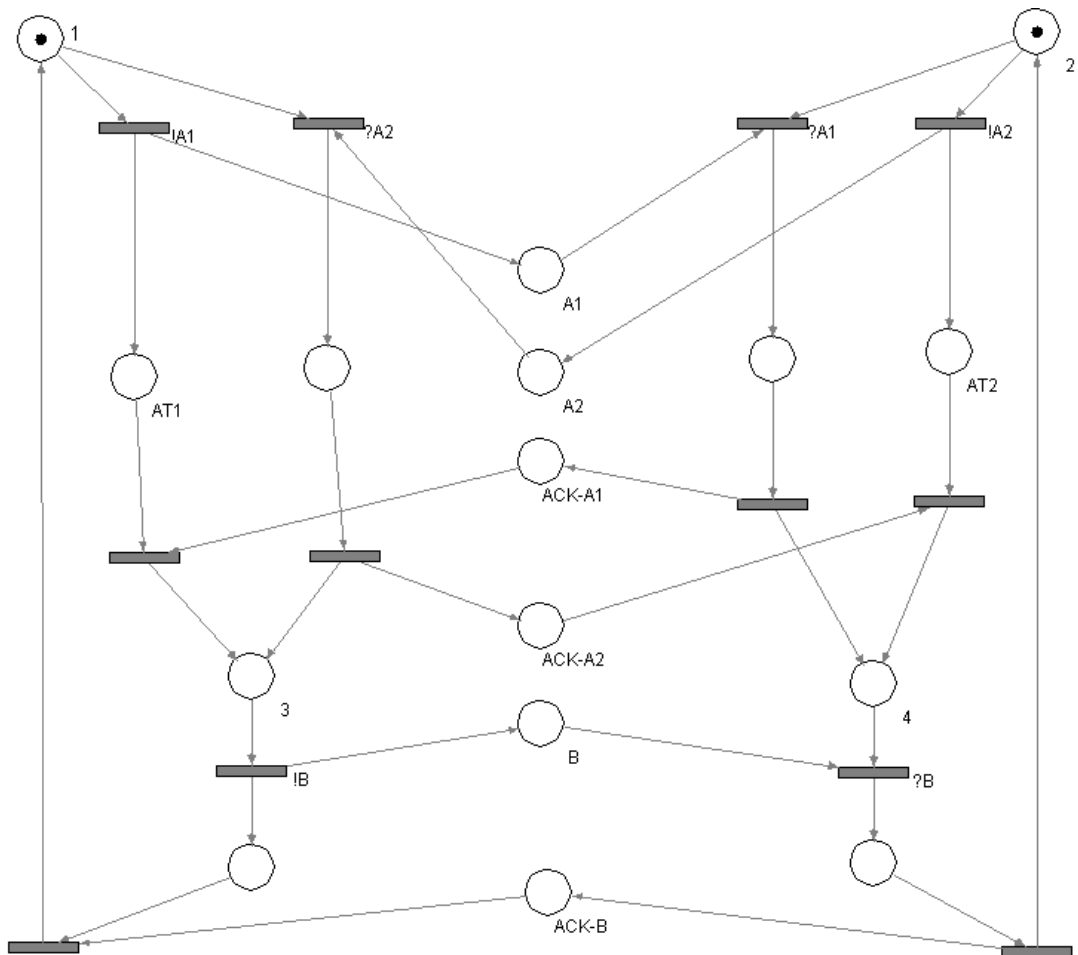
### Solution B : envoi de signal



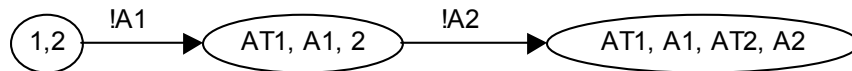
La section de graphe d'accessibilité ci-dessous exprime le fait que les places A1 et B peuvent recevoir un nombre arbitrairement élevé de jetons. Ce comportement est susceptible d'entraîner un dépassement des ressources.



### Solution C : appel-réponse



La section de graphe d'accessibilité ci-dessous exprime un comportement qui amène un blocage. En effet, les tirs, dans un ordre quelconque de !A1 et !A2, conduisent au marquage des places AT1, A1, AT2 et A2. Dans cet état, aucune transition n'est tirable.



### Synthèse

Le tableau suivant résume les trois comportements distincts des modèles présentés.

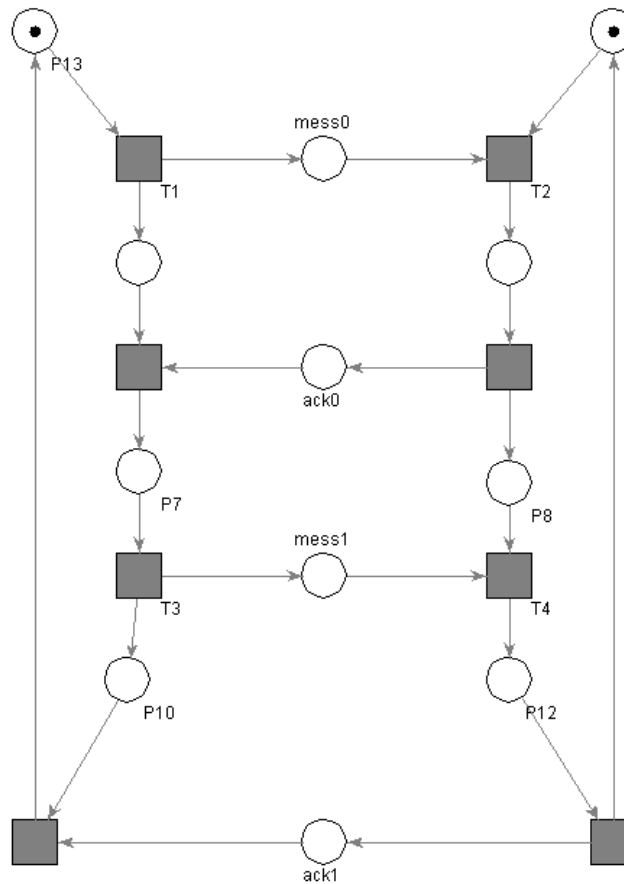
Solution	Comportement
A	borné et vivant
B	non borné et vivant
C	borné et bloquant

On remarquera qu'aucun des modèles ne prend en considération la possibilité de perdre un message lors de la communication. C'est ce qui nous amène à étudier un protocole différent, celui du bit alterné.

### 3.3 Protocole du bit alterné (BA)

Dans ce protocole, les données sont émises en leur associant un bit de contrôle, successivement de valeur 0 et 1 : la première donnée sera émise en lui associant un bit 0, puis la suivante un bit 1, et ainsi de suite. C'est une solution simple mais non performante. Elle permet cependant d'éliminer les mauvais fonctionnement liés aux pertes et aux duplications de messages. Voici une première modélisation de ce protocole.

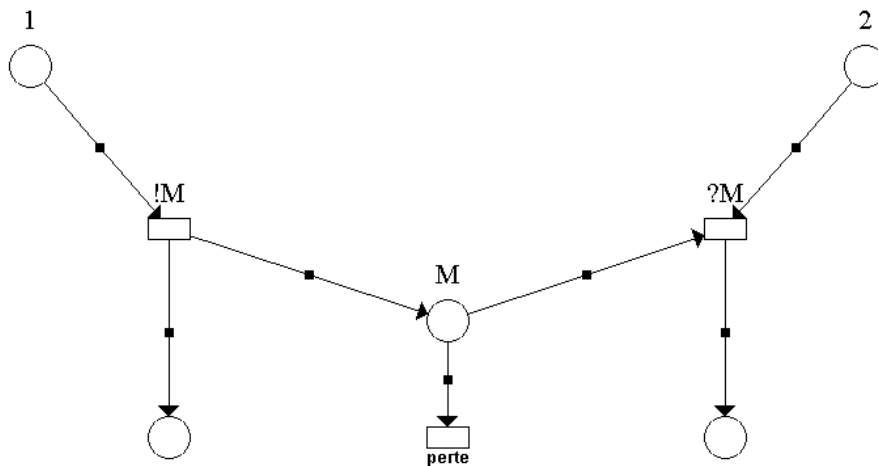
#### Exemple 8



Quelques remarques sur cette modélisation :

- ◇ Le processus émetteur envoie la première donnée avec le bit 0 (mess0), attend la réception de la réponse liée à ce bit (ack0), envoie la deuxième donnée avec le bit 1 (mess1), attend la réception de la réponse liée à ce bit (ack1), envoie la troisième donnée avec le bit 0 (mess0), etc.
- ◇ Le processus de réception attend la première donnée associée au bit 0 (mess0), envoie la réponse liée à ce bit (ack0), attend la seconde donnée associée au bit 1 (mess1), envoie la réponse liée à ce bit (ack1), attend la troisième donnée associée au bit 0 (mess0), etc.
- ◇ C'est au travers une place partagée que l'interconnexion entre l'envoi et la réception des données est modélisée.

Nous avons remarqué qu'aucun des modèles de la section 3.1 ne prenait en considération la possibilité de perdre un message lors de la communication. Cette perte est facile à modéliser, en voici le modèle de base.



Il s'agit en fait de la modélisation de ce qu'on appelle « un datagramme », un mécanisme de base dans les protocoles courants : Ethernet, IP, UDP. Il exprime le simple envoi de message unique, sans aucun autre contrôle.

Incorporons maintenant ce modèle de base dans la modélisation complète du protocole (voir l'exemple 9). D'abord, la perte des données et la perte des réponses (ack) se modélisent de façon identique puisqu'il n'est pas possible pour l'émetteur de savoir si c'est la donnée ou la réponse qui a été perdue.

Après la perte d'une donnée associée au bit « i », une place « perte-i » indique que la donnée a été perdue. Cette place est essentielle afin de savoir si cette donnée perdue doit être renvoyée avec la même valeur du bit de contrôle.

Le RdP de l'exemple 9 est en fait une généralisation de celui de l'exemple 8. On y a ajouté, dans un premier temps, les places « perte-i » et les six transitions « pd-i », « pa-i » et « red-i ». Les transitions « pd-i » expriment la perte de la donnée « i ». Les transitions « pa-i » expriment la perte de la réponse « i ». Finalement, les transitions « red-i » expriment la demande d'en renvoi de la donnée « i ». Mais cette modélisation (non illustrée ici) n'est pas suffisante car existeraient alors deux cas de blocages. On y ajoute donc quatre places supplémentaires « don-i » et « rea-i » pour obtenir la modélisation finale illustrée à l'exemple 9.

- ◇ « rea0 » permet de représenter la réception d'un nouvel envoi (duplication) de la donnée « don0 » (déjà reçue), à la suite de la perte de la réponse « ack0 », avec renvoi de cette réponse. On remarquera que cette nouvelle donnée, reçues deux fois, ne doit pas être mémorisée la deuxième fois.
- ◇ « rea1 » permet de représenter de façon similaire la réception d'une duplication de la donnée « don1 », à la suite de la perte de la réponse « ack1 ».

### Exemple 9

