# Réseaux de Petri

Etienne Renault

Avril 2015

https://www.lrde.epita.fr/~renault/teaching/imc/

# Comment modéliser un système ?

La modélisation de systèmes concurents soulève de nombreux problèmes :

- Comment exprimer la concurrence ?
  - Processus / Threads
- Comment modéliser les communications ?
  - Synchrones / Asynchrones
- Comment modéliser les systèmes avec un nombre d'états non-fini ?
- Comment valider cette modélisation ?

Les automates synchronisé ne sont pas suffisants.

Problème des communications asynchrones par exemple !

# Carl Adam Petri (1/3)

- Né le 12 juillet 1926 à Leipzig et mort le 2 juillet 2010 à Siegburg
- Mathématicien et Informaticien Allemand
- Centres d'intérêts : calculs parallèles, calculs distribués, systèmes complexes, workflow, ...
- 1962 : Thèse intitulée "Communication par les automates"

# Carl Adam Petri (2/3)

## Point de départ !

Soit $f$ une fonction récursive arbitraire prenant un argument $n$. Quel est l'espace mémoire nécessaire pour calculer $f(n)$ ?

- Les pré-requis mémoires pour calculer $n$ peuvent être beaucoup plus importants que ceux pour calculer $n - 1$
- On alloue des ressources et on calcule $f(n)$.
  - Si $f(n)$ termine cela signifie que l'on a alloué suffisamment de ressources
  - Sinon, il faut allouer plus de ressources et recommencer.

## Idée

Peut-on construire un système de calcul dans lequel un composant peut être rajouté sans relancer le système ?

# Carl Adam Petri (3/3)

Petri a proposé l'ajout de la standardisation des I/O dans ALGOL comme moyen de communication entre des processus séquentiels. Cette proposition a été refusée et c'est un des éléments qui conduira à l'abandon d'ALGOL

Petri a proposé un formalisme (pendant sa thèse) permettant la composition et l'allocation de ressources appelé : **Petri net**

- ce formalisme a été étendu à de nombreuses reprises
- ce formalisme est à la base de nombreux formalismes de compositions de logiciels (diagrammes d'activité UML, ...)
- l'un des premiers travaux sur une approche basée sur un développement dirigé par les modèles !

# Les réseaux de Petri

Un réseau de Petri est un 5-uplet $\langle P, T, W^-, W^+, M_0 \rangle$ tq :

- $P$ est un ensemble fini non-vide de places
- $T$ est un ensemble fini de transitions
- $W^- : P \times T \to \mathbb{N}$ la fonction d'incidence avant
- $W^+ : P \times T \to \mathbb{N}$ la fonction d'incidence arrière
- $M_0 \in \mathbb{N}^P$ est le marquage initial du réseau (il indique le nombre de *jetons* dans une place donné)

$W^-(p, t)$ est la précondition associée à la transition $t$ et à la place $p$. Elle défini le nombre de marques (jetons) qui doivent être présentes dans $p$ pour que $t$ soit franchissable. Si $t$ est franchie, ce nombre de jeton est retiré de $p$.

$W^+(p, t)$ est la postcondition associée à la transition $t$ et à la place $p$. Elle défini le nombre de marques (jetons) qui sont généré dans $p$ lorsque $t$ est franchie. de $p$.

# Un exemple simple (1/2)

- Les places sont représentées par des ronds

  *place_name*

- Les transitions sont représentées par des rectangles

  *transition_name*

- Les jetons sont représentés par des billes noires à l'intérieur des places (ici trois jetons)

  *place_name*

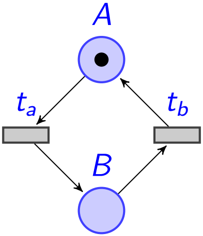
Tous les éléments sont reliés par des flêches ($\rightarrow$) mais l'on ne peut pas relier des transitions entre-elles ou deux places entre-elles !

# Un exemple simple (2/2)

- $P = \{A, B\}$
- $T = \{t_a, t_b\}$
- $M_0(A) = 1,\ M_0(B) = 0$
-

| $W^-$ | $t_a$ | $t_b$ |
|---|---|---|
| $A$ | 1 | 0 |
| $B$ | 0 | 1 |

| $W^+$ | $t_a$ | $t_b$ |
|---|---|---|
| $A$ | 0 | 1 |
| $B$ | 1 | 0 |



# Faire circuler les jetons

Une transition $t$ est dite franchissable pour un marquage $M \in \mathbb{N}^P$ ssi $\forall p \in P, M(p) \geq W^-(p,t)$.

Le marque $M'$ alors obtenu est défini par :

$$\forall p \in P, M'(p) = M(p) - W^-(p,t) + W^-(p,t)$$

On note $M[t > M'$, ce qui signifie que $t$ est franchissable pour le marquage $M$ et atteint le marquage $M'$

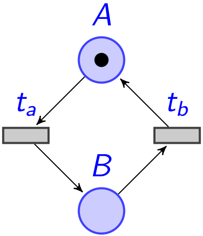
# Faire circuler les jetons

Une transition $t$ est dite franchissable pour un marquage $M \in \mathbb{N}^P$ ssi $\forall p \in P, M(p) \geq W^-(p,t)$.

Le marque $M'$ alors obtenu est défini par :

$$\forall p \in P, M'(p) = M(p) - W^-(p,t) + W^-(p,t)$$

On note $M[t > M'$, ce qui signifie que $t$ est franchissable pour le marquage $M$ et atteint le marquage $M'$



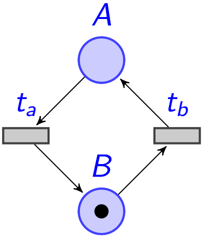
# Faire circuler les jetons

Une transition $t$ est dite franchissable pour un marquage $M \in \mathbb{N}^P$ ssi $\forall p \in P, M(p) \geq W^{-}(p, t)$.

Le marque $M'$ alors obtenu est défini par :

$$\forall p \in P, M'(p) = M(p) - W^{-}(p, t) + W^{-}(p, t)$$

On note $M[t > M'$, ce qui signifie que $t$ est franchissable pour le marquage $M$ et atteint le marquage $M'$



# Quelques remarques

## Franchissement et concurrence

Les règles d'évolution (ou de franchissement) garantissent :

- l'atomicité d'un franchissement
- le fait que les notions de séquentialité, de non-déterminisme ou de concurrence sont à égalité puisque toutes sont vues comme des schémas particulier de causalité.

# Quelques remarques

## Franchissement et concurrence

Les règles d'évolution (ou de franchissement) garantissent :

- l'atomicité d'un franchissement
- le fait que les notions de séquentialité, de non-déterminisme ou de concurrence sont à égalité puisque toutes sont vues comme des schémas particulier de causalité.

## Matrice d'incidence

Les matrices d'incidence avant et arrières sont parfois fusionnées en une matrice d'incidence $W : P \times T \to \mathbb{N}$, définie par :

$$\forall (p, t) \in P \times T, W(p, t) = W^{+}(p, t) - W^{-}(p, t)$$

# Quelques remarques

## Franchissement et concurrence

Les règles d'évolution (ou de franchissement) garantissent :

- l'atomicité d'un franchissement
- le fait que les notions de séquentialité, de non-déterminisme ou de concurrence sont à égalité puisque toutes sont vues comme des schémas particulier de causalité.

## Matrice d'incidence

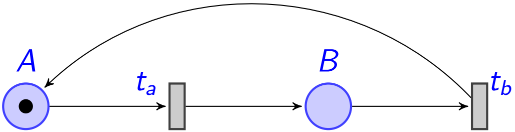
Les matrices d'incidence avant et arrières sont parfois fusionnées en une matrice d'incidence $W : P \times T \to \mathbb{N}$, définie par :

$$\forall (p,t) \in P \times T, W(p,t) = W^+(p,t) - W^-(p,t)$$

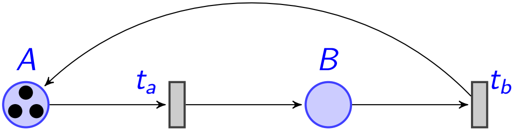
Attention, donner seulement $W$ masque les boucles alors que les matrices $W^-$ et $W^+$ ne les masquent pas !

# Quelques exemples

- Un système composé d'un processus

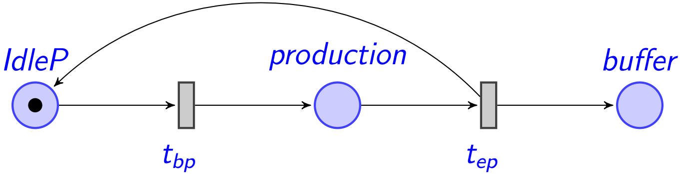



- Le même système mais avec trois processus identiques



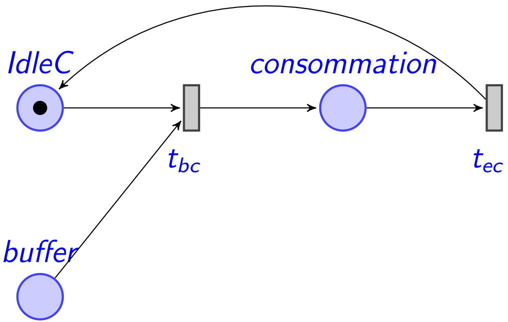
# Producteur / Consommateur(1/3)

- **Producteur** : à chaque fois que la transition $t_{ep}$ est franchie un nouvel objet est déposé dans le buffer.

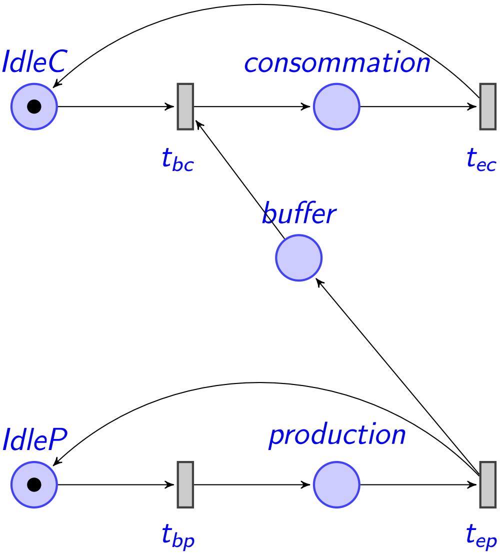


# Producteur / Consommateur(2/3)

- **Consommation** : la transition $t_{bc}$ ne peut être franchie que s'il y a un objet dans le buffer.



# Producteur / Consommateur (3/3)

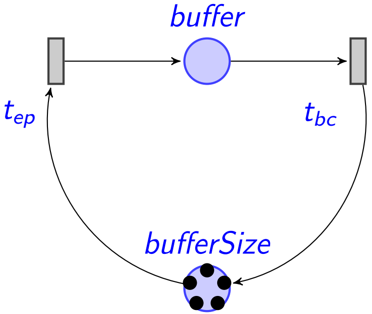


# Rajouter un composant !

Comment limiter la taille du buffer pour qu'il ne contienne que 5 éléments au maximum ?

# Rajouter un composant !

Comment limiter la taille du buffer pour qu'il ne contienne que 5 éléments au maximum ?



# Marquages accessibles

Soit $R = \langle P, T, W^-, W^+, M_0 \rangle$ un réseau de Petri. L'ensemble des marquages accessibles de ce réseau, noté $Acc(R)$ est défini par :

$$Acc(R) = \{M \in \mathbb{N}^P | \exists s \in T^* \text{ tq } M_0[s > M\}$$

# Marquages accessibles

Soit $R = \langle P, T, W^-, W^+, M_0 \rangle$ un réseau de Petri. L'ensemble des marquages accessibles de ce réseau, noté $Acc(R)$ est défini par :

$$Acc(R) = \{M \in \mathbb{N}^P | \exists s \in T^* \text{ tq } M_0[s > M\}$$

Le graphe des marquages accessibles est le graphe orienté et valué dont les nœuds sont les éléments de $Acc(R)$ et est tel qu'il existe un arc valué par $t \in T$ de $M$ vers $M'$ ssi $M[t > M'$

# Marquages accessibles

Soit $R = \langle P, T, W^-, W^+, M_0 \rangle$ un réseau de Petri. L'ensemble des marquages accessibles de ce réseau, noté $Acc(R)$ est défini par :

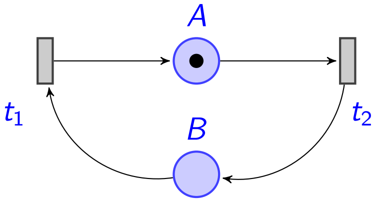
$$Acc(R) = \{M \in \mathbb{N}^P | \exists s \in T^* \text{ tq } M_0[s > M\}$$

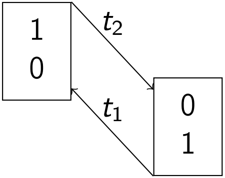
Le graphe des marquages accessibles est le graphe orienté et valué dont les nœuds sont les éléments de $Acc(R)$ et est tel qu'il existe un arc valué par $t \in T$ de $M$ vers $M'$ ssi $M[t > M'$

Ce graphe n'est pas nécessairement fini, et même lorsqu'il est fini, la taille de ce graphe est souvent trop importante pour analyser des propriétés dessus.

# Exemples de marquages accessibles

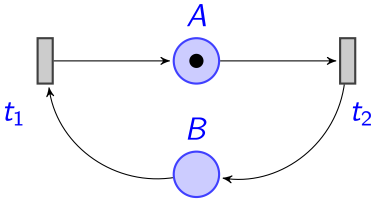
- Fini

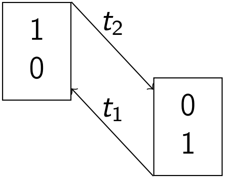


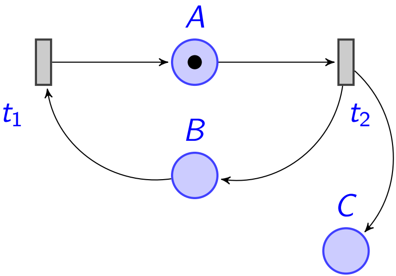
# Exemples de marquages accessibles

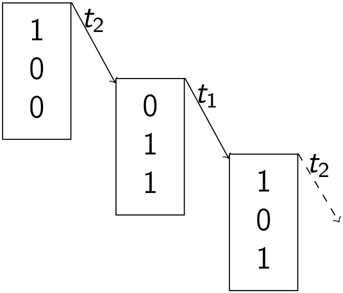
- Fini




- Infini



# Structure d'un réseau de Petri

## Analyse structurelle

L'analyse de la structure du réseau de Petri permet d'obtenir des informations : réseau borné, pas de blocage, exclusion mutuelle, et dans certains cas la vivacité du réseau.

Les invariants permettent d'analyser un modèle sans avoir développer tout ou partie du graphe des marquages accessibles.

# Places et réseaux bornés

Soit $R = \langle P, T, W^-, W^+, M_0 \rangle$ un réseau de Petri. Une place $p \in P$ est dite $k$-bornée ssi $\forall M \in Acc(R), M(p) \leq k$.

Un réseau de Petri est dit $k$-borné si toutes ses places sont $k$-bornées.

Un réseau $k$-borné possède un graphe des marquages accessibles (aussi appelé graphe d'accessibilité) fini.

Attention ! si un réseau est $k$-borné cela ne signifie pas qu'il n'y a que $k$ jetons qui circule dans le réseau !

# Invariants de place

Soit $X \in \mathbb{Z}^P$ un vecteur entier sur les places. Si $W(t)$ désigne la $t^{ieme}$ colonne de la matrice d'incidence $W$, alors l'équation de changement d'état peut s'écrire :

$$^{T}X.M' = {}^{T}X.M + {}^{T}X.W(t)$$

Si $^{T}X.W(t) = 0$ alors il vient que $^{T}X.M' = {}^{T}X.M$, et en particulier $^{T}X.M' = {}^{T}X.M_0$ : c'est à dire que la somme des marques contenues dans l'ensemble des places reste constante moyennant une pondération après le franchissement de $t$.

Les solutions entières $X$ de l'équation $^{T}X.W = 0$ seront donc les invariants du réseau, aussi appelé flots.

Lorsque tous les coefficients d'un flot sont positifs, ce flot est appelé semi-flot

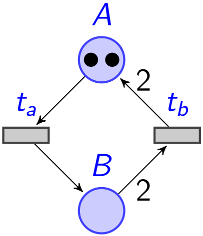
# Invariants – Exemple

- $P = \{A, B\}$
- $T = \{t_a, t_b\}$
- $M_0 = [2, 0]$
- 

| $W^-$ | $t_a$ | $t_b$ |
|---|---|---|
| $A$ | 1 | 0 |
| $B$ | 0 | 2 |

| $W^+$ | $t_a$ | $t_b$ |
|---|---|---|
| $A$ | 0 | 2 |
| $B$ | 1 | 0 |

| $W$ | $t_a$ | $t_b$ |
|---|---|---|
| $A$ | -1 | 2 |
| $B$ | 1 | -2 |



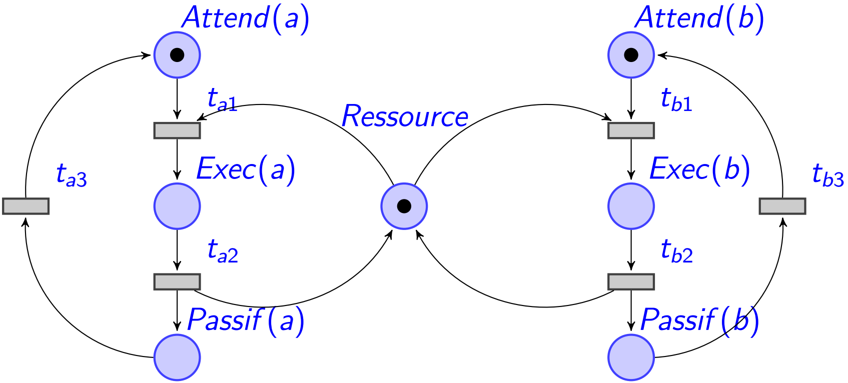

Si ${}^TX = [1, 1]$, alors ${}^TX.W = 0$.
$A + B$ est donc un invariant du réseau

# Familles génératrices

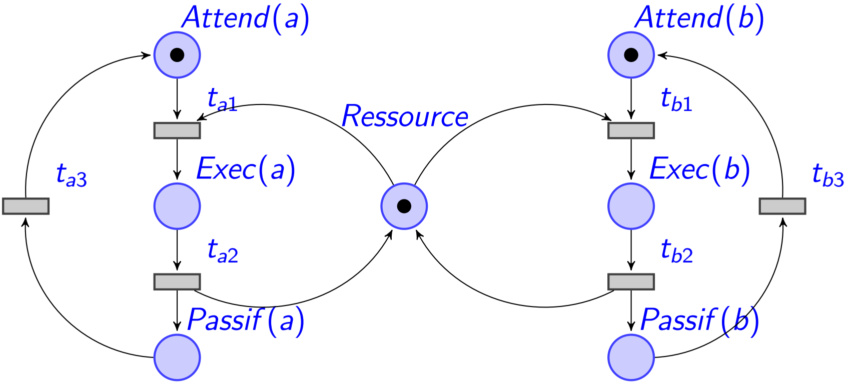
Une famille génératrice de semi-flots est le plus petit ensemble de semi-flots permettant d'exprimer, par combinaison linéaire, tous les semi-flots d'un réseau de Petri.

Les flots peuvent se calculer par l'algorithme de Gauss tandis que les semi-flots peuvent se calculer par l'algorithme de Farkas.

# Exclusion mutuelle

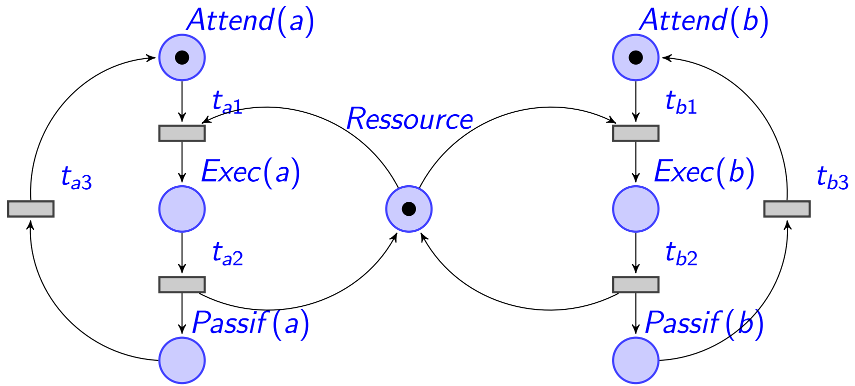


# Exclusion mutuelle




## Semi-Flots

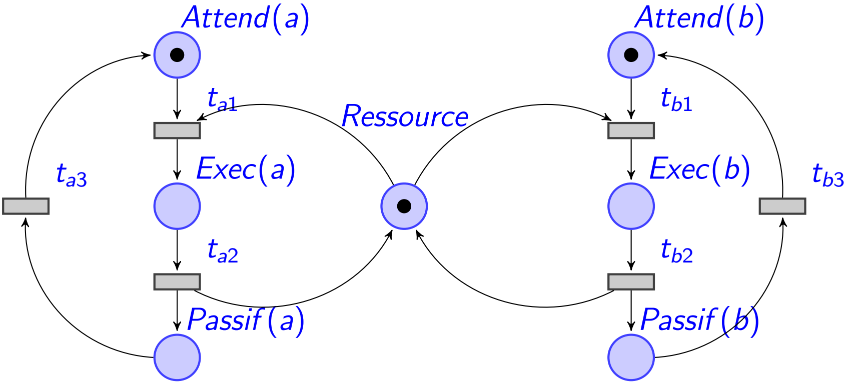
# Exclusion mutuelle




## Semi-Flots

- $Attend(a) + Exec(a) + Passif(a) = 1$

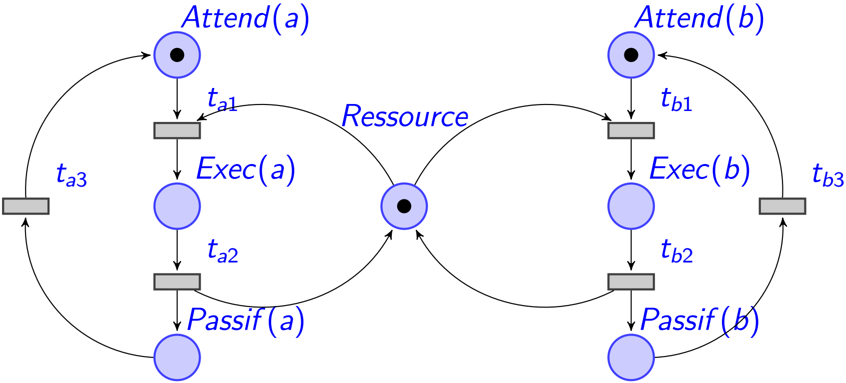
# Exclusion mutuelle




## Semi-Flots

- $Attend(a) + Exec(a) + Passif(a) = 1$
- $Attend(b) + Exec(b) + Passif(b) = 1$

# Exclusion mutuelle




## Semi-Flots

- $Attend(a) + Exec(a) + Passif(a) = 1$
- $Attend(b) + Exec(b) + Passif(b) = 1$
- $Exec(a) + Exec(b) + Ressource = 1$

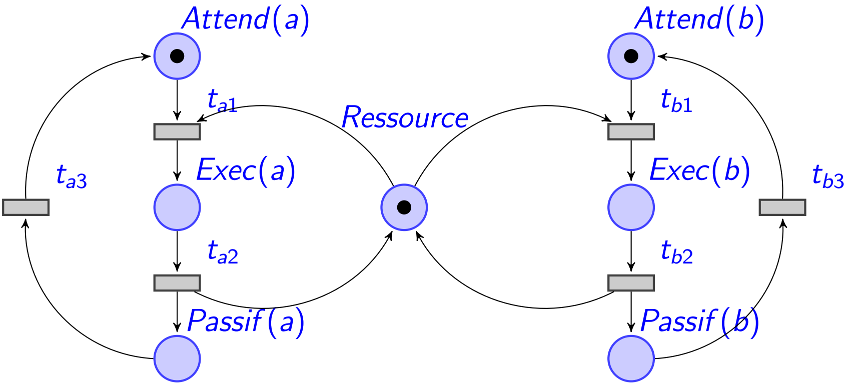
# Interprétation des Semi-Flots

Soit $F$ une famille génératrice de semi-flots. Certains éléments $f \in F$, permettent de décomposer le réseau de Petri en processus et en ressources

- un **processus** : est un ensemble *séquentiel* lié à la notion de machine à état ;
- une **ressource** : est une place qui n'est contenue dans aucun processus.

Si $|f| = |F|$, il n'y a qu'une interprétation possible, i.e. un unique processus.
Sinon il peut y avoir plusieurs interprétations possibles, chacune correspondant à un point de vue sémantique.

# Exclusion mutuelle (interprétation)




## Deux interprétations possibles

- 2 processus, 1 ressource
- 1 processus, 4 ressources

# Propriétés de base

Un réseau de Petri $R$ est dit :

- **pseudo-vivant** ssi $\forall M \in Acc(R), \exists t \in T$ t.q. $M[t >$

  *Autrement dit, depuis le marquage initial, il existe toujours une transition qui peut être franchie.*

- **quasi-vivant** ssi $\forall t \in T, \exists s \in T^*$ t.q. $M_0[s > M'[t >$

  *Autrement dit, depuis le marquage initial, toutes les transitions peuvent être franchies au moins une fois.*

- **vivant** ssi $\forall M \in Acc(R), \forall t \in T, \exists s \in T^*$ t.q. $M[s > M'[t >$

  *Autrement dit, quel que soit l'évolution du réseau depuis le marquage initial, toutes les transitions sont franchissables.*

# Réduction des réseaux de Petri

Lors de la modélisation certains états/transitions *inutiles* (du point de vue de la vérification de certaines propriétés) peuvent être introduits.

Comment supprimer ces états/transitions ?

# Réduction des réseaux de Petri

Lors de la modélisation certains états/transitions *inutiles* (du point de vue de la vérification de certaines propriétés) peuvent être introduits.

Comment supprimer ces états/transitions ?

La réduction d'un réseau de Petri est une transformation qui réduit la taille du réseau tout en préservant un ensemble de propriétés. Nous nous intéressons ici à deux types de propriétés :

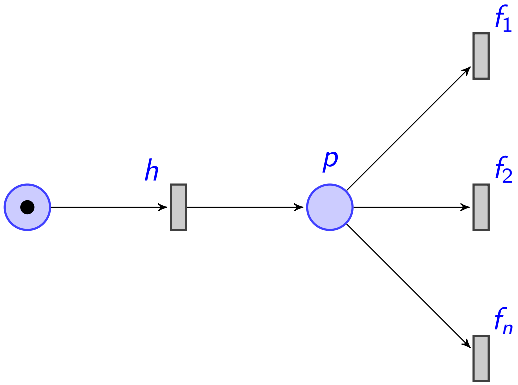
- l'agglomération de transitions (pré ou post)
- la suppression de places

# Transitions pré-agglomérables

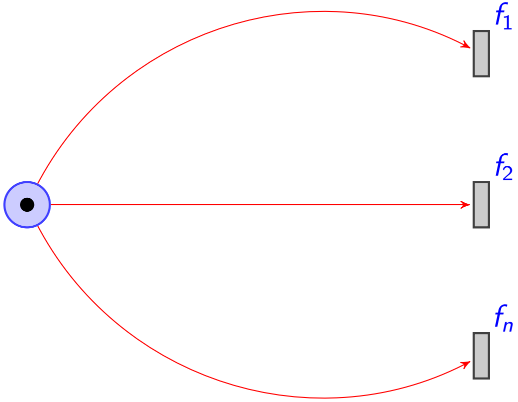
Soit $R = \langle P, T, W^-, W^+, M_0 \rangle$ un réseau de Petri. Un ensemble de transitions $F$ est pré-agglomérable avec une transition $h \notin F$ ssi les conditions suivantes sont vérifiées :

1. Il existe une place $p$ modélisant un état intermédiaire entre le franchissement de $h$ et d'une transitions de $F$ :
   1. $M_0(p) = 0$
   2. ${}^\bullet p = \{h\}$ et $p^\bullet = F$
   3. $W^+(p, h) = 1$
   4. $\forall f \in F, W^-(p, f) = 1$
2. $h$ ne produit que des marques dans $p$ : $h^\bullet = \{p\}$
3. $h$ n'est en conflit avec aucune autre transition : $\forall q \in {}^\bullet h, q^\bullet = \{h\}$
4. $h$ a au moins une pré-condition : ${}^\bullet h \neq \emptyset$

# Transitions pré-agglomérables – exemple



# Transitions pré-agglomérables – exemple

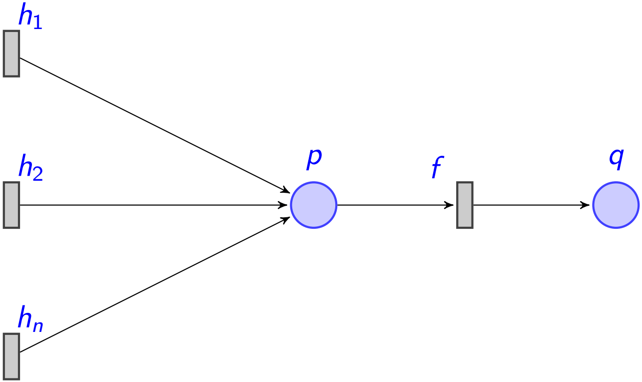


# Transitions post-agglomérables

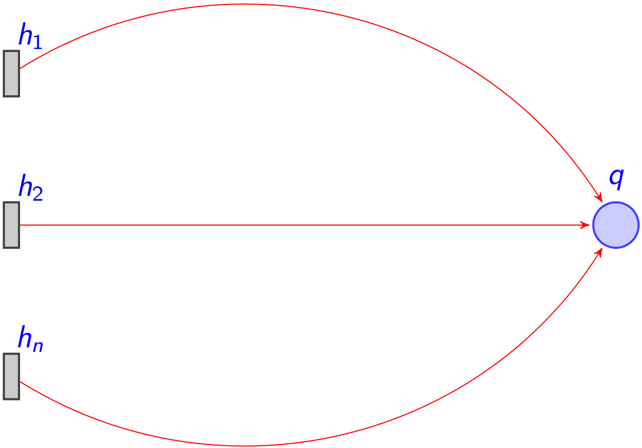
Soit $R = \langle P, T, W^-, W^+, M_0 \rangle$ un réseau de Petri. Un ensemble de transitions $F$ est post-agglomérable avec un ensemble de transitions H avec $H \cap F = \emptyset$ ssi les conditions suivantes sont vérifiées :

1. Il existe une place $p$ modélisant un état intermédiaire entre le franchissement d'une transition de $H$ et d'une transitions de $F$ :
   1. $M_0(p) = 0$
   2. ${}^\bullet p = H$ et $p^\bullet = F$
   3. $\forall h \in H, W^+(p, h) = 1$
   4. $\forall f \in F, W^-(p, f) = 1$
2. Les transitions de $F$ n'ont pas d'autres pré-conditions que $p$ : ${}^\bullet F = \{p\}$
3. Il existe une transition $f \in F$ ayant une post-condition : $F^\bullet \neq \emptyset$

# Transitions post-agglomérables – exemple



# Transitions post-agglomérables – exemple



# Agglomérations - résumé

## Pré-agglomération

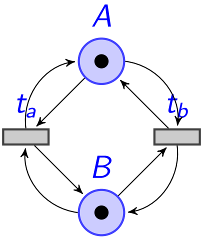
Cette opération repose sur l'idée que l'on peut dans toute séquence de franchissement comportant une occurrence de transition $h$ puis d'une autre transition $f$, retarder le franchissement de $h$ pour le confondre avec celui de f.

## Post-agglomération

La post-agglomération repose donc sur l'idée que l'on peut, dans toute séquence contenant une occurrence d'une transition $h$ puis d'une transition $f$, avancer le franchissement de $f$ pour le confondre avec celui de $h$.

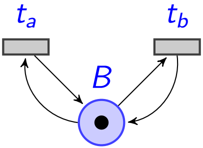
# Places implicites

Une place implicite est une place qui n'est jamais un obstacle au franchissement de transitions.



# Places implicites

Une place implicite est une place qui n'est jamais un obstacle au franchissement de transitions.



# Autres réductions

## Fusion de places doublées

Ce type de transformation vise à fusionner des places lorsque les marques qui apparaissent dans l'une ne peuvent être confondues avec celles qui apparaissent dans l'autre.

## Fusion de places équivalentes

Deux places équivalentes sont deux places telles que les suites d'évolutions possibles quand l'une est marqué sont les mêmes que quand l'autre est marquée.

# Réseau de Petri de haut niveau (1/2)

Les réseaux de Petri constituent un cadre théorique simple et puissant pour l'étude de la concurrence ...

# Réseau de Petri de haut niveau (1/2)

Les réseaux de Petri constituent un cadre théorique simple et puissant pour l'étude de la concurrence ...

...mais mettent en avant les contrôles aux détriment des données ! Il est alors facile d'analyser et de modéliser un système mais il est difficile d'intégrer dans la phase de conception les donées échangées.

# Réseau de Petri de haut niveau (1/2)

Les réseaux de Petri constituent un cadre théorique simple et puissant pour l'étude de la concurrence ...

...mais mettent en avant les contrôles aux détriment des données ! Il est alors facile d'analyser et de modéliser un système mais il est difficile d'intégrer dans la phase de conception les donées échangées.

## Les abréviations

Afin de prendre en compte ce besoin sans modifier la sémantique des réseaux de Petri, différentes abréviations ont été définies.

Une abréviation de réseau est obtenue en attachant de nouvelles informations aux noeuds et aux arcs et en définissant une sémantique de fonctionnement qui tient compte de ces paramètres.

# Réseau de Petri de haut niveau (2/2)

## Les réseaux colorés

Ces réseaux sont caractérisés par une sémantique fonctionnelle simple : une information de couleur (ou de type) est associée aux jetons et aux franchissements, et les valuation des arcs sont des fonctions de couleurs qui précisent le nombre et la couleur des jetons consommés/produits.

## Les Well-formed Nets

Ces réseaux constituent une abréviation fonctionnelle et paramétrée qui restreint les fonctions de couleurs à des compositions élémentaires (identité, successeurs, et diffusion). et qui impose que les couleurs manipulées soient des tuples de valeurs prises dans des *classes*.

# FIACRE

**process** node [pred :none, suc : none](leader : bool) **is**
**states** cs , waiting , idle
init **if** leader **then** **to** cs **else** **to** idle **end**
**from** idle **to** waiting
**from** waiting pred ; **to** cs
**from** cs suc ; **to** idle

**component** main **is**
**port** port01 , port12 , port20 : none **par**
**node** [ port20 , port01 ] ( true )
|| **node** [port01 , port12] (false)
|| **node** [port12 , port20] (false)
**end**
main

# Conclusion

Les réseaux de Petri constituent un formalisme :

- simple à comprendre et manipuler
- avec une sémantique concurrente forte
- des propriétés structurelles permettant d'analyser directement le réseau sans passer (pour certains cas) par l'utilisation de logique temporelle
- idéal pour la modélisation mais qui reste parfois 'un peu trop haut niveaux'