Modélisation avec les Réseaux de Petri

Modélisation avec les Réseaux de Petri

J. Christian Attiogbé

Janvier 2020

Nantes Université

Motivations

Exercices

- Construire un logiciel pour contrôler simultanément plusieurs robots coopérant dans une usine/maison/entreprise.
- Construire un logiciel pour gérer l'interaction entre plusieurs joueurs dans un logiciel de jeu.
- Construire un logiciel pour gérer l'interaction entre des applications qui partagent une ressource.
- Construire un logiciel qui gère l'interaction entre des applications qui, à distance, les unes produisent des données et les autres, consomment les données.

Méthodes

Quelles méthodes ? quels modèles ?
Quelles technologies ? quels langages ? quelles plateformes ?

Motivations

- A travers cet enseignement, vous allez acquérir les connaissances de base pour comprendre, modéliser et développer les systèmes/logiciels répartis.
- acquérir les concepts et mécanismes de base de la répartition,
- mettre en œuvre les compétences acquises sur des exemples et études de cas.

Objectifs du cours

Objectifs

Ce cours a pour but de donner aux étudiants les méthodes et techniques de base pour la modélisation, la conception, et la réalisation de systèmes et logiciels répartis (*distributed systems*).

Moyens

Formation encadrée : CM 6h (4*1h20) - TD/TP 17,5h (7 sem., 2*1h20/sem)
Prérequis : Systèmes M3101, Réseaux M3102
Fil conducteur : applications réelles à réaliser au fil des TDs

Intervenants

Christian Attiogbé (CM, TD) & Loïg Jezequel (TD)

Plan de la suite (adaptation selon contraintes 2020-21)

Modélisation avec les réseaux de C. A. PETRI

Introduction

- Carl Adam Petri, 1962
- Une notation mathématique et graphique
- Fondée sur une bonne théorie
- Plusieurs extensions et outils logiciels pour les analyser
- Modélisation de systèmes asynchrones, concurrents, distribués, non-déterministes,
- Utilisable à différentes étapes de construction de logiciels : analyse, modélisation, simulation, développement (synthèse)
- Utilisation dans de nombreux domaines : protocoles, systèmes critiques, systèmes d'exploitation, ...

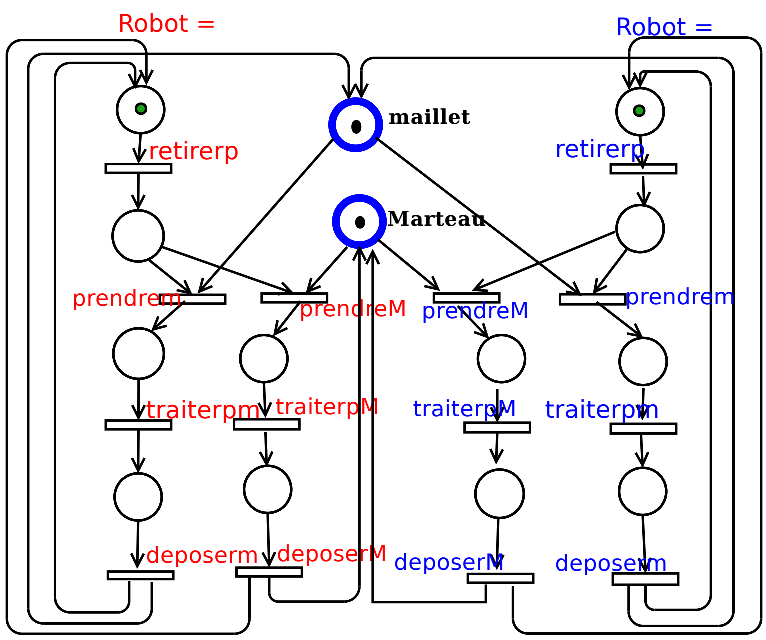
Introduction

Le réseau de Petri comme un outil de l'ingénieur : modélisation et étude du fonctionnement d'un système (avant sa construction).

Les réseaux de Petri permettent de modéliser+analyser des systèmes discrets ; c'est à dire ceux dont les variables d'entrée, sortie et état sont discrètes.
(et si ce n'est pas discret, comment fait-on ? d'autres outils...)

Exemple :
Contôle dans les ateliers de production.
Modélisons le comportement de deux robots dans une usine ;
les deux robots usinent des pièces en se servant d'outils communs.
Plusieurs problèmes : accès concurrent, synchronisation, blocage, ...

Introduction : un exemple de concurrence



Caractéristiques des réseaux de Petri

- Spécification et étude des systèmes concurrents (communication, synchronisation)
- Abstraction sur les comportements, et les états
- Mode de synchronisation : synchrone, asynchrone
- Mode de composition : via le partage de places/transitions, (pas de hiérarchie)
- Modèles sémantiques : opérationnelle, axiomatique (algébrique)

Concepts fondamentaux

- Ensemble de Places : P
- Ensemble de transitions : T
- Arcs entrant/sortant des transitions et des places
- Jetons (dans des places)
- Marquage (des places) : M_i
  On marque les places avec des jetons
- Fonctionnement du réseau : franchissement des transitions et donc, changement de l'état global (marquage).

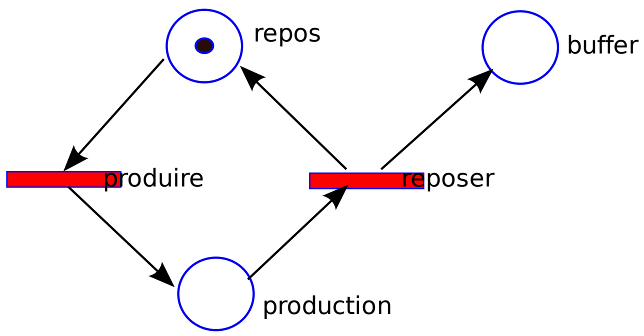
Etude empirique du fonctionnement des R. de Petri

Exemples de modèles élémentaires à l'aide de RdP

- Modélisation d'un système producteur/consommateur
- Modélisation d'une chaîne d'assemblage avec des robots ...

Exemple : Producteur/Consomateur

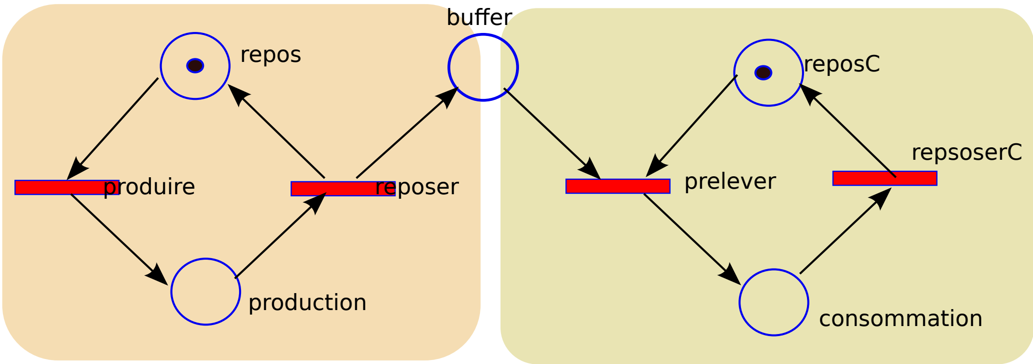
Modélisation du comportement d'un producteur :



$$P = \{repos,\ production,\ buffer\}$$
$$T = \{produire,\ reposer\}$$

Exemples : fonctionnement

Modélisation de la composition producteur et consommateur :



Définition formelle d'un RdP et notations

Un réseau de Petri (R) est un quadruplet $(P,\ T,\ Pre,\ Post)$ où :

- P : un ensemble fini de places (avec $\mid P \mid \ = m$, le cardinal de P),
- T : un ensemble fini de transitions, disjoint de P, $(\mid T \mid \ = n)$
- $Pre\ :\ P \times T \rightarrow N$ une application d'incidence avant (une matrice) : les places entrant dans une transition
- $Post\ :\ P \times T \rightarrow N$ une application d'incidence arrière (une matrice) : les places sortant d'une transition

Notations étendues : $Pre(*, t)$, $Pre(t)$, $Post(*, t)$, $Post(t)$

Il reste le marquage (où sont les jetons) !

Exemple

Soient
$P = \{1, 2, 3, 4, 5\}$ et $T = \{a, b, c, d, e\}$.

On peut définir/construire le réseau de Petri (avec Pre et Post) suivant :

Pre (places avant les transitions)

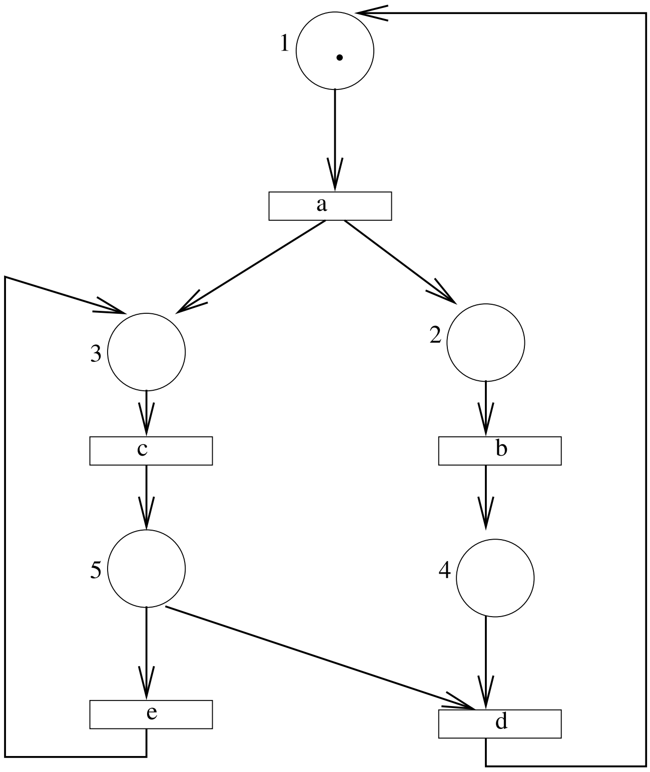
Pre	a	b	c	d	e
1	1	0	0	0	0
2	0	1	0	0	0
3	0	0	1	0	0
4	0	0	0	1	0
5	0	0	0	1	1

Post (après les transitions) :

Post	a	b	c	d	e
1	0	0	0	1	0
2	1	0	0	0	0
3	1	0	0	0	1
4	0	1	0	0	0
5	0	0	1	0	0

Il reste le marquage (où sont les jetons) !

Graphe du réseau



Marquage des réseaux

Un réseau marqué est le couple $N = (R, \mu)$ formé de :

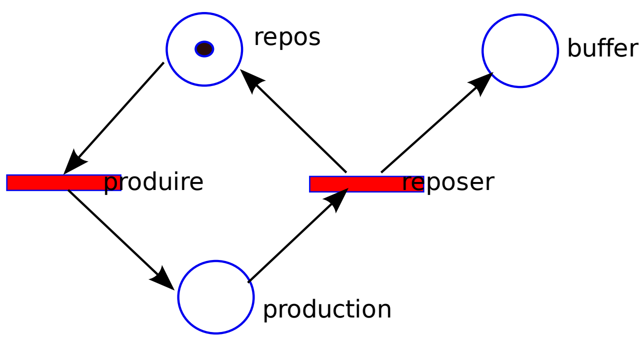
- un réseau R et
- une application (fonction totale) $\mu : P \rightarrow N$.
  $\mu(p)$ est le marquage de la place p,
  on dit aussi le nombre de marques contenues dans p.

Jeton : indique le marquage de chaque place.

Le **marquage initial est noté** M_O

On peut limiter le marquage des places à 1 ou à un entier k.
Réseau k-borné : $\forall p.\mu(p) \leq k$ propriétés sur le modèle !

Marquage des places



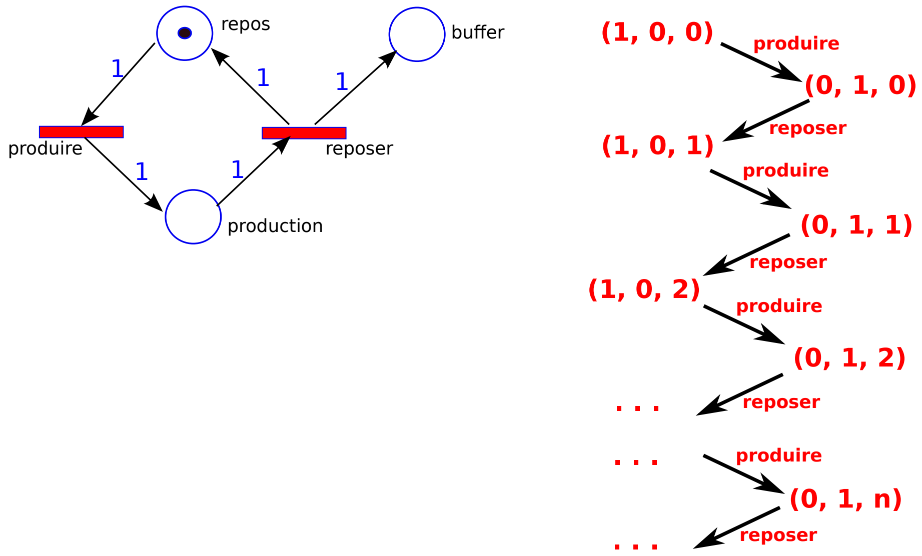
$$\mu(repos) = 1$$
$$\mu(production) = 0$$
$$\mu(buffer) = 0$$

$$M_i = (\mu(repos), \mu(production), \mu(buffer))$$

$$M_i = (1, 0, 0)$$

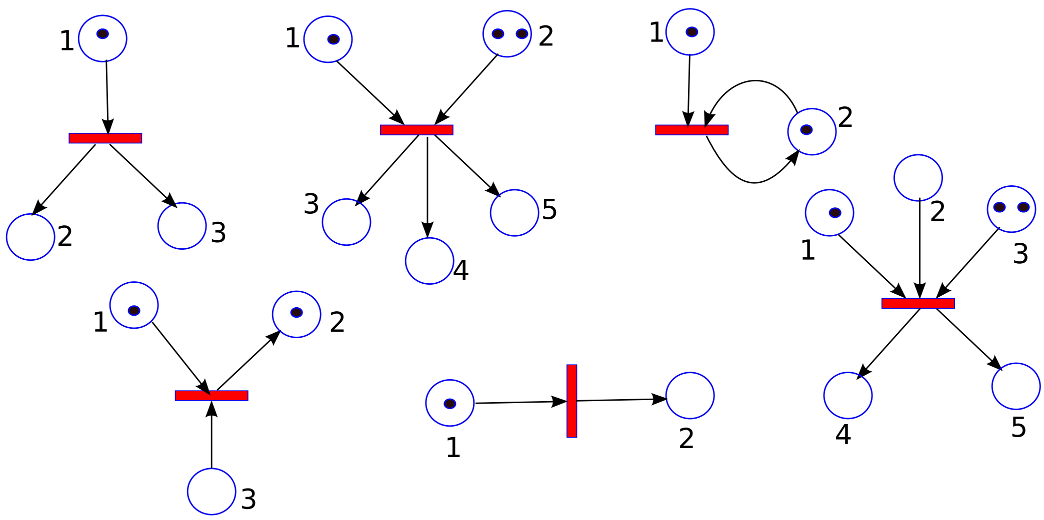
Graphe de Marquage

Les marquages des places constituent des états ou noeuds ; toutes les transitions possibles à partir d'un noeud engendrent des transitions d'un noeud à un autre noeud dans le graphe des marquages.



Fonctionnement et propriétés des réseaux de Petri

- Franchissement des transitions (non-déterminisme, synchronisation)
  Franchissable $(M \rightarrow t)$? quel est le marquage M' résultant ?



Fonctionnement des réseaux de Petri (règles sémantiques)

- Une transition t est franchissable (ou tirable) (*enabled*) lorsqu'il y a au moins un jeton dans chacune de ses places en entrée ($Pre(t)$).
  $M \rightarrow t \sim \forall p.\mu(p) \geq Pre(p,t)$ **ou bien** $M \geq Pre(t)$
- Une transition franchissable est franchie (instantanément) ou non franchie.
- Lorsque plusieurs transitions sont franchissables,
  une d'entre elles (de façon non-déterministe) est franchie.
- Lorsqu'une transition est franchie,
  on enlève un jeton de chacune de ses places en entrée et
  on ajoute un jeton dans chacune de ses places en sortie :
  modification du marquage. $M' = M - Pre + Post$
  $M \rightarrow^t M'$

Chaque arc a un poids de 1 par défaut.

Fonctionnement du réseau : graphe et propriétés

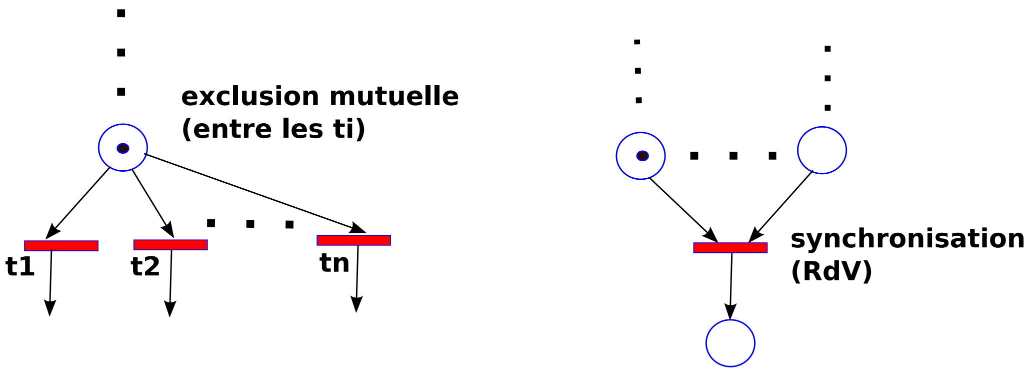
On étudie le modèle à partir du graphe de marquage (réseau borné).
On construit des chemins ($\sigma = t_1.t_2....$) comme une suite de transitions.

- Marquage et vivacité :
  Transition vivace (vivante) = il y a toujours un chemin qui y passe ;
  Réseau vivace = le chemin arrivant à tout noeud sans arc sortant contient au moins chaque transition
  Réseau sans blocage = à partir d'un noeud il y a toujours un arc sortant.
- Interblocage : aucune action possible à partir d'un noeud atteignable.
- ...

On peut avoir des réseaux non bornés.

Schémas fondamentaux dans les réseaux de Petri

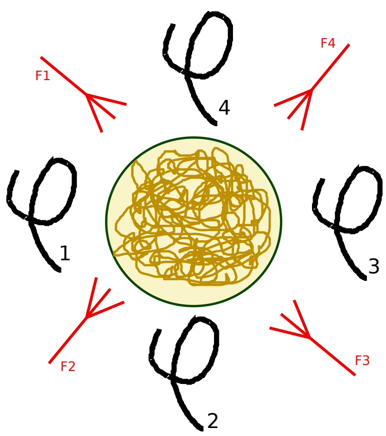
- Franchissement de transitions : exclusion mutuelle, synchronisation (RdV)



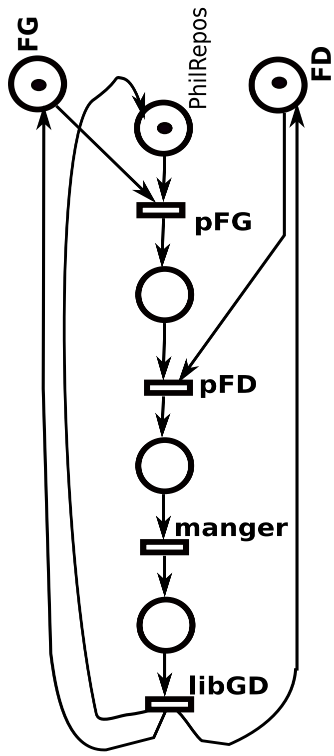
Fonctionnement du réseau : graphe et propriétés

Exemples de modèles génériques

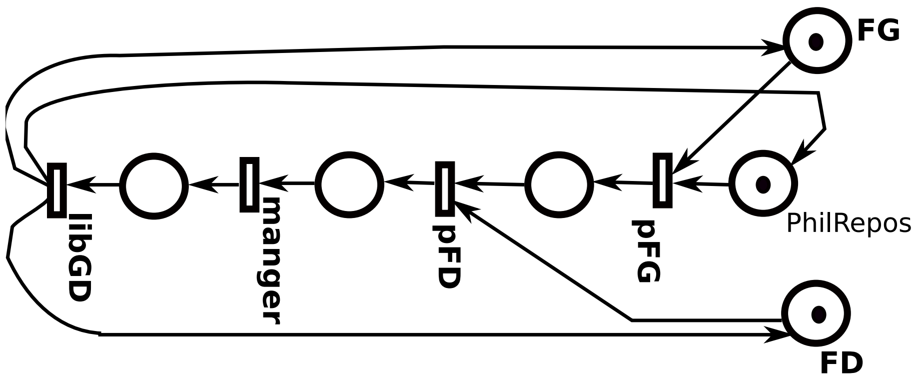
- Les philosophes
- Lecteurs/rédacteurs
- ...



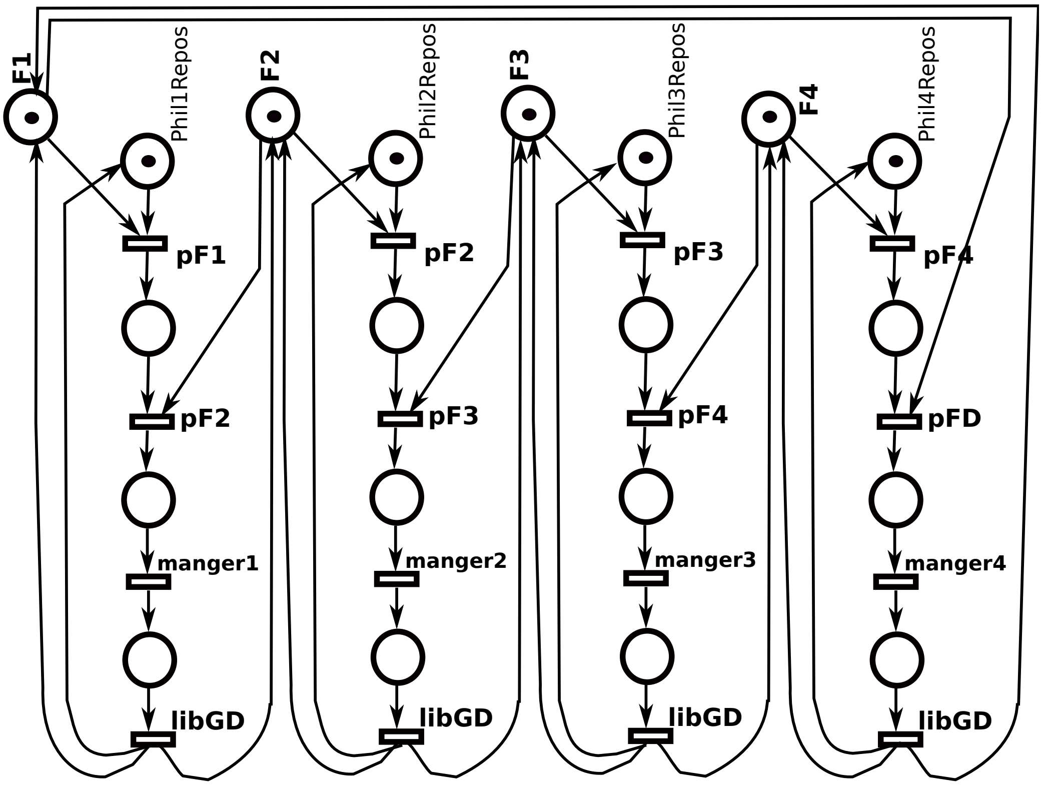
Modélisation du comportement - philosophe



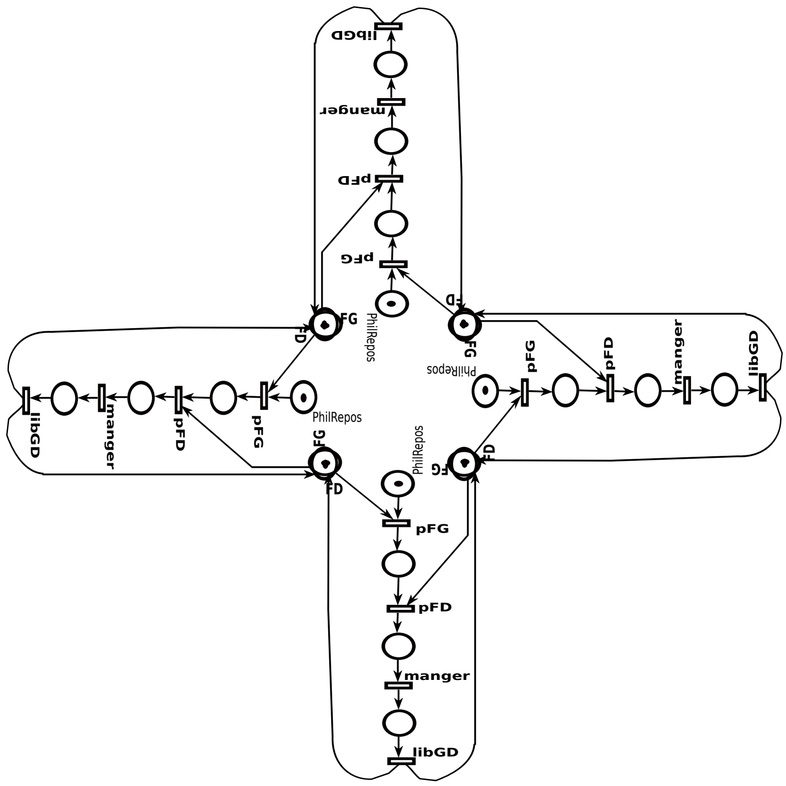
Modélisation du comportement - philosophe



Modélisation du comportement - 4 philosophes



Modélisation du comportement - 4 philosophes



Problème de ce modèle : interblocage

Une situation (configuration / état du système) :

- chaque philosophe prend une fourchette (G) !
- Chaque philosophe attend la deuxième fourchette (D)

Ils sont tous en attente l'un de l'autre !
interblocage : propriété indésirable pour certaines applications

Analyse de la source du problème : allocations indépendantes de ressources partagées.
Solution générique : tout-ou-rien (les ressources)
A mettre en œuvre dans les applications réparties

Graphe associé à un réseau

Le graphe (P, T, Γ, V) associé au réseau $R = (P, T, Pre, Post)$ est défini par :

- $\forall p \in P . \Gamma_p(p) = \{t \in T \mid Pre(p, t) > 0\}$
  (transitions atteintes par chaque p)
- $\forall t \in T . \Gamma_t(t) = \{p \in P \mid Post(p, t) > 0\}$
  (places atteintes par chaque t)
- $\forall p \in P, \forall t \in T. V(p, t) = Pre(p, t)$ et $V(t, p) = Post(p, t)$
  (valuation des arcs)

$Pre(*, t)$ ou $Pre(t)$ indique la colonne de la transition t

Graphe = structure interne du RdP, propice à l'analyse formelle (algos).

Exemple

Soient $P = \{1, 2, 3, 4, 5\}$ et $T = \{a, b, c, d, e\}$.

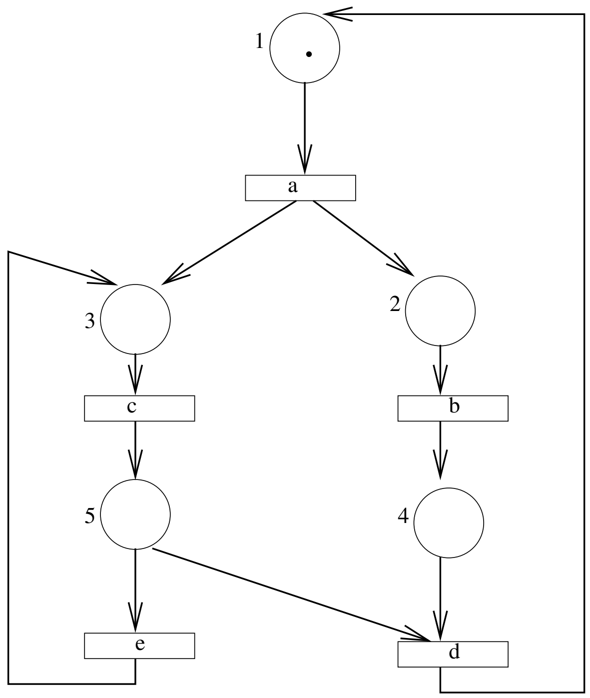
et

Pre	a	b	c	d	e
1	1	0	0	0	0
2	0	1	0	0	0
3	0	0	1	0	0
4	0	0	0	1	0
5	0	0	0	1	1

Post	a	b	c	d	e
1	0	0	0	1	0
2	1	0	0	0	0
3	1	0	0	0	1
4	0	1	0	0	0
5	0	0	1	0	0

soit $M_0 = (1\ 0\ 0\ 0\ 0)$ le marquage initial.
On construit le réseau suivant avec M_0 :

Graphe du réseau



$$\Gamma_p(1) = \{a\}$$
$$\Gamma_t(a) = \{2, 3\}$$
$$\ldots$$

Entrée et sortie d'une transition

- On appelle entrées d'une transition t : les places $\Gamma^{-1}(t)$ et
- On appelle sorties d'une transition t : les places de $\Gamma(t)$.

- On appelle entrées d'une place p : les transitions $\Gamma^{-1}(p)$ et
- sorties d'une place p : les transitions de $\Gamma(p)$.

Exercice

On donne Γ_p et Γ_t telles que : (on utilisera Γ_t pour les transitions et Γ_p pour les places et Γ abusivement pour les deux cas)

$$\begin{array}{lcl} \Gamma_p(1) & = & \{a\} \\ \Gamma_p(2) & = & \{b\} \\ \Gamma_p(3) & = & \{c\} \\ \Gamma_p(4) & = & \{d\} \\ \Gamma_p(5) & = & \{d,\ e\} \end{array} \qquad \begin{array}{lcl} \Gamma_t(a) & = & \{2,\ 3\} \\ \Gamma_t(b) & = & \{4\} \\ \Gamma_t(c) & = & \{5\} \\ \Gamma_t(d) & = & \{1\} \\ \Gamma_t(e) & = & \{3\} \end{array}$$

Construisez le graphe du réseau associé à cette description.

Graphe de marquage

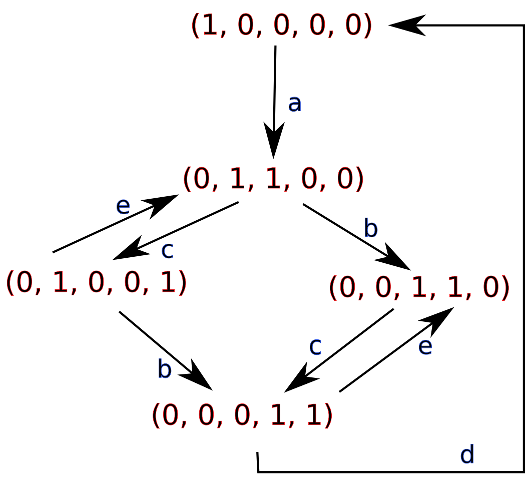
On construit le graphe sur l'ensemble des marquages accessibles noté $A(R,\ M)$ (ou espace d'états) du réseau.
Un état est un marquage.
Si les places du réseau sont $p_1, p_2, \ldots, p_{|P|}$,
un marquage est de la forme $(\mu(p_1),\ \mu(p_2),\ \ldots, \mu(p_{|P|}))$.

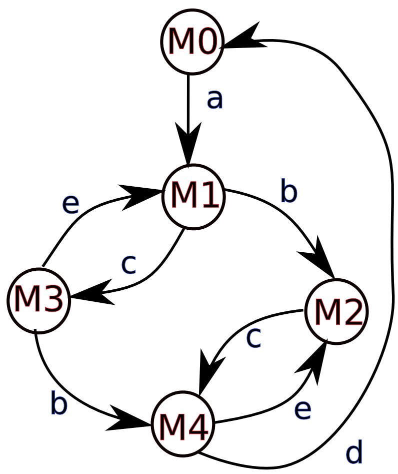
Le graphe des marquages est noté $G(R,\ M)$.
Ses sommets sont les éléments de $A(R,\ M)$.
Ses arcs sont étiquetés par t_i tel que $t_i\ \in\ T$.

Soient M_i des sommets, $M_1 \ \rightarrow^t \ M_2$ ssi :

- $M_1,\ M_2\ \in\ A(R,\ M)$,
- $t\ \in\ T$,
- M_2 est accessible par t à partir de M_1

Graphe de marquage





Chaque noeud peut être abstrait/nommé (M_i).
Le graphe peut devenir gigantesque !
(à construire/exploiter avec des algorithmes)

Algorithme de calcul du graphe de marquage

```
MAccessibles := {marquageInitial} /* ensemble de marquages*/
AExplorer := MAcessibles /* ensemble des marquag. à explorer*/
TANTQUE AExplorer /= {}  FAIRE
 Mc := prendre un marquage de AExplorer /* marquage courant*/
 AExplorer := AExplorer - {Mc} // différence ensembliste
 POUR toutes les transitions t de Mc FAIRE
   SI Mc >= Pre(t)  ALORS /* t est franchissable */
     M' := Mc + Post(t) - Pre(t) /* un marquage accessible */
     SI M' /: MAccessibles  ALORS
      Accessibles := Accessibles ∪ {M'} // union ensembliste
      AExplorer := AExplorer ∪ {M'}
     FINSI
   FINSI
 FINPOUR
FINFAIRE
```


Propriétés des réseaux

- k-Borné (1-borné = binaire / booléen)
- Transition vivante (apparait dans toute trace ; franchissable à partir de tout marquage) : sur le chemin des marquages accessibles à partir de tout marquage courant.
- Transition quasi-vivante : existe/franchissable sur au moins un chemin. A partir d'un marquage donné, on peut franchir la transition.
- Réseau vivant : toutes ses transitions sont vivantes (très important pour certains systèmes industriels ; ne se dégradent pas...).
- Réseau réinitialisable : on repasse par le marquage initial
- Réseau avec blocage (mortel) ; à partir d'un certain marquage ; plus aucune transition vivante :-(

Pondération des arcs, arcs inhibiteurs

Arcs pondérés

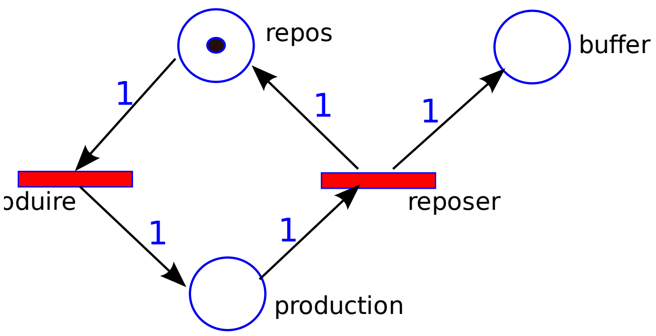
- Arcs pondérés par défaut avec 1
- Les arcs peuvent avoir un poids > 1

Exemple d'utilisation :

- Lecteurs/rédacteur ; on veut autoriser n lectures ; mais exclusion entre un rédacteur et les lecteurs ; rédaction en exclusion avec les lecteurs.

Pondération des arcs

Par défaut les arcs ont un poids de 1

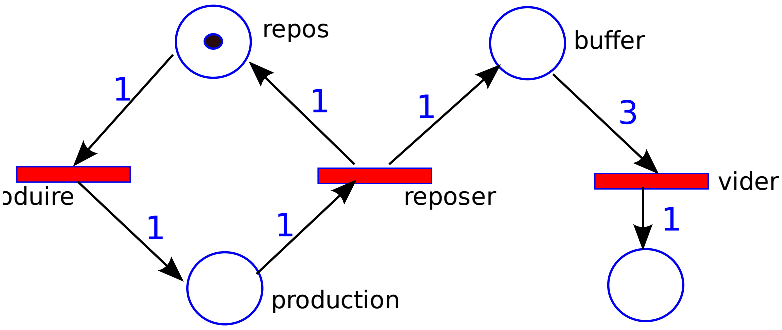


Rappel : Lorsqu'une transition est franchie, on enlève **un** jeton de chacune de ses places en entrée et on ajoute **un** jeton dans chacune de ses places en sortie : modification du marquage. $M' = M - Pre + Post$

Mais cela correspond au poids des arcs . . . on généralise maintenant

Pondération des arcs

Exercice : consommation 3 par 3 de ce qui est produit :



La transition vidée est franchissable s'il y a aumoins 3 jetons dans le buffer. Pour la mise à jour, on retire 3 jetons !

Pondération des arcs

Exercice : Modélisation de la composition des comportements des processus lecteurs/rédacteurs. N ressources sont disponibles pour la lecture ou la rédaction.

Exigences de l'application:
- un seul processus rédacteur à la fois,
- 1 à N processus lecteurs peuvent lire simultanément,
- On veut assurer l'exclusion entre lecture et rédaction.

La solution, c'est avec les arcs pondérés.

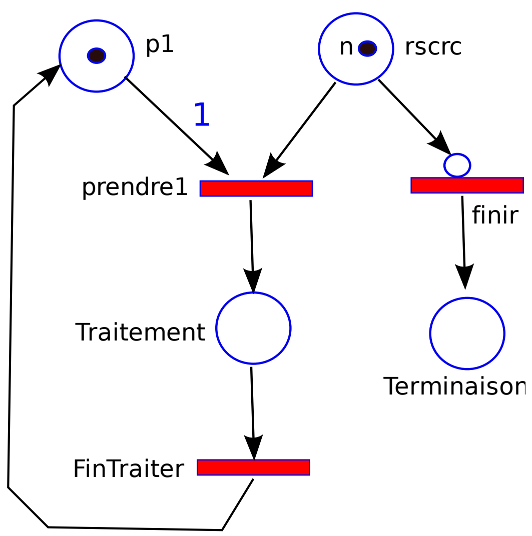
Arcs inhibiteurs

Arcs inhibiteurs : permettent des franchissements spécifiques.
Un RdP à arcs inhibiteurs est une variante des RdP, où on peut avoir une transition spécifique franchissable lorsque le marquage de la place entrante est 0.

Exercice : patients/médecin (au tableau)

Arcs inhibiteurs

Modélisation de la détection de fin d'un traitement...
(ici on modélise le fait que l'appli s'arrête (finir) quand il n'y a plus de patients)



La transition finir est franchissable quand le nombre de jetons de la place rscrc est 0.

Les limites des réseaux de Petri

- Manque de modularité
  - pas de hiérarchisation
  - pas de composition explicite
- Explosion du modèle (pour la modélisation de grands systèmes)
- Prise en compte de la modélisation des données
- ...

Les extensions des réseaux de Petri

Il existe de nombreuses extensions pour les réseaux de Petri

- Réseaux de Petri avec des données (lang. de données élaboré)
  Réseaux de Petri colorés (le jetons ont une couleur/type) ;
  Réseaux de Petri hiérarchiques ;
  Réseaux de Petri avec temps ;
  Réseaux de Petri stochastiques ;
  . . .
- HLPN (*High-Level Petri Nets*, réseaux de Petri avec prédicats, etc)
- ...

Références

De nombreuses références existent :

- *Understanding Petri nets. Modeling techniques, analysis methods, case studies.* W. Reisig,
- Génie logiciel, Réseaux de Petri, Racloz et Buchs
- Les réseaux de Petri, Un outil de modélisation, Annie Choquet-Geniet, Dunod
- ...
- `www.scholarpedia.org/article/Petri_net`
- Applications `https://cs.au.dk/cpnets/industrial-use/software/`