Introduction aux Réseaux de Petri

François Vernadat – vernadat@laas.fr

http://homepages.laas.fr/francois/RdP

INSA-DGEI – LAAS-CNRS

2015

Outline

Type d'Applications/Systèmes visés

Programme "Classique" (séquentiel)

Caractéristiques :

- Termine
- Retourne un résultat
- Données +/- complexes mais contrôle séquentiel ("simple")

Exemples :
"Fonctions" (tri, code, calcule, traitement image/signal, transforme/compilate, ...)

Ingrédients pour les exprimer : fonctions/procédures, structures de données, structures de contrôle conditionnelles/itératives, ...

Langages : Impératif, fonctionnel, objet, logique, ...

Correction = correction partielle + terminaison
correction partielle : *si le programme termine il fournit un résultat correct*
terminaison : *le programme termine toujours*

Système Concurrents/Réactifs (>70)

Caractéristiques :

- Σ composé par des processus indépendants
- Communication, Synchronisation, ...
- "partage" de ressources
- Σ ne retourne pas de résultat
- Terminaison en général indésirée (deadlock)
- Données +/- simples mais contrôle concurrent ("compliqué")

Exemples :
Protocoles de communication, Contrôle/Commande, Σ-Embarqués, ...

Ingrédients pour les exprimer : état/transition, non-déterminisme, parallélisme, synchronisation, communication, ...

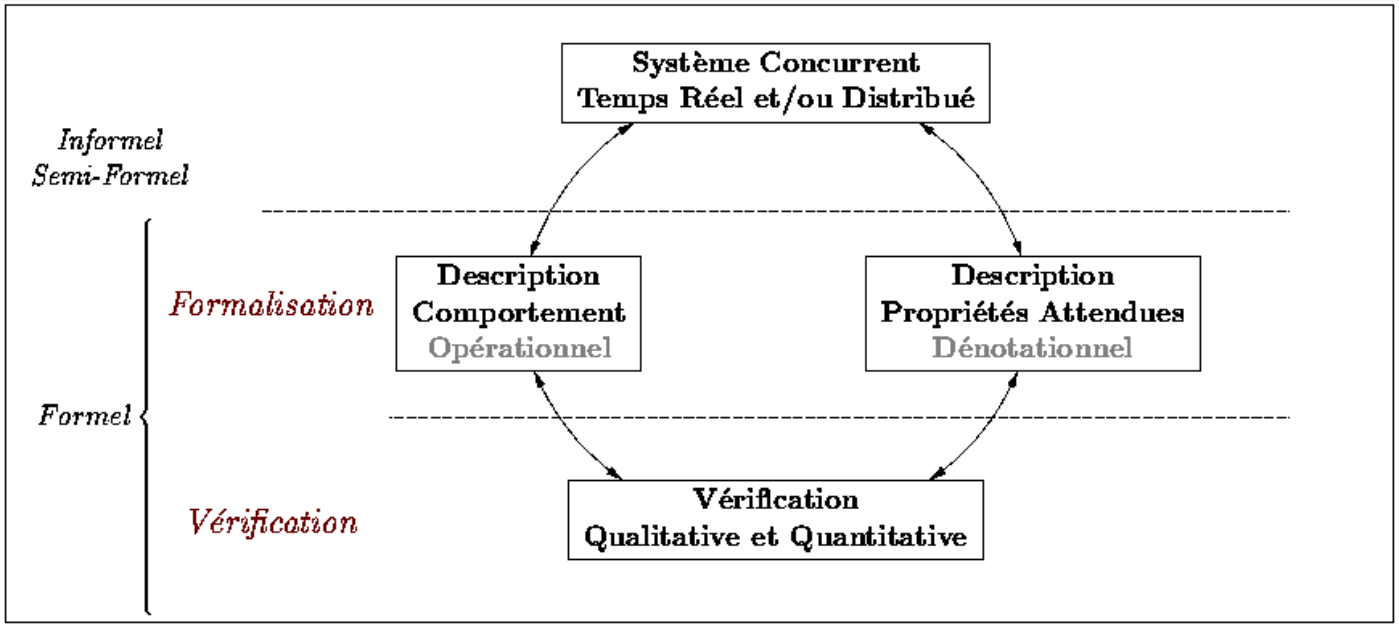
Formalismes : Machines à états/Automates communicants, Réseaux de Petri, Algèbres de processus ...

Correction = sûreté, vivacité, équité, ...
Nécessite le plus souvent une analyse exhaustive de l'espace d'états
$\mapsto$ model-checking

Model-Checking

What is model checking ? – Amir Pnueli 2000

"Model checking is the method by which a desired behavioral property of a reactive system is verified over a given system (the model) through exhaustive enumeration (explicit or implicit) of all the states reachable by the system and the behaviors that traverse through them."



Ingrédients

$\mathcal{M}$: Modèle (formel) du comportement (**issu de** Rdp, Automates, Alg.Proc,..)
ϕ : Modèle (formel) des propriétés (formalisme dédié)
MC : Algorithme de vérification permettant de décider si $\mathcal{M}$ satisfait ϕ?
nb : Formel est une condition requise pour envisager l'automaticité

Propriétés types des systèmes concurrents/réactifs

- **Accessibilité**
  - Une certaine situation peut être atteinte
  - Chaque action peut avoir lieu
- **Sûreté**
  - Quelque chose de mauvais n'arrive jamais

  Jamais plus de 2 processus en section critique

  Jamais de débordement de buffers

  La barrière est fermée lorsqu'un train est sur le passage à niv eau
- **Vivacité** (*absence de famine*)
  - Quelque chose de bon finit par arriver

  Un processus en attente d'une ressource finir par l'obtenir

  Un philosophe affamé finira par manger
- **Equité(s)**

  - Si une action peut avoir lieu une infinité de fois alors elle aura lieu une (infinité) de fois, ...
- Vivacité sous condition d'équité

  S'il n'y a pas une infinité de pertes, tout message émis - perdu et retransmis - finira par arriver (Alternating Bit Protocol)

Réseaux de Petri

Historique

1962 Thèse de Carl Adam Petri (RFA)

1970 Travail important (France & RFA)

1980 Conférence Annuelle (Springer-Verlag)

Diffusion Mondiale : Universitaire & "Industrielle"
http ://www.daimi.aau.dk/PetriNets/ :

Intérêt

- Formalisme Rigoureux (sémantique précise)
- Expressivité
  + Compact (vs machines à états)
  + Concepts : //, Coopération, Compétition, Synchronisation, ...
  + Dualité Etat/Evénement
- Possibilité d'Analyse : Structurelle, Exhaustive
- Représentation Graphique
- Diverses extentions : temporel (temps réel), stochastique (performance), ...
- Nombreux outils (certains couplés à des outils/formalismes industriels)

Sur ce cours ....

Objectifs

Courte introduction aux réseaux de Petri et à la modélisation/analyse de systèmes concurrents

- un peu de théorie en CM
- un peu de pratique en "TP" libre
- URL A completer

Organisation : 6H CM + travail personnel + exam 1H avec doc

Contenu : Introduction aux Réseaux Place/Transition

- Définitions & Concepts de base
- Enumération, Propriétés
- Décidabilité de la finitude de l'espace d'états accessibles
- Invariants (Analyse Structurelle)

Bibliographie

G.W Brams (ouvrage collectif)
Réseaux de Petri : Théorie et Pratique (2 tomes) - Masson -1980

W. Reisig
Petri Nets. An Introduction Springer-Verlag EATCS 1985

C. Reutenauer
Aspect Mathématiques des Réseaux de Petri
Etudes et recherches en informatique Masson - 1989

K. Jensen
Coloured Petri Nets - Springer-Verlag EATCS 1992

W. Reisig
Elements od Distributed Algortithms : Modeling and Analysis with Petri Nets Springer-Verlag 1998

M. Diaz (Editeur)
Les Réseaux de Petri. Modèles fondamentaux,
Hermes Science, Traité IC2, 2001

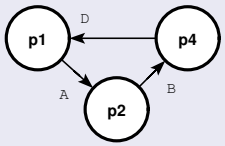
M. Diaz (Editeur)
Petri Nets. Fundamental Models, Verification and Applications,
ISTE & Wiley, N 978-1-84821-079-0, 2009

Outline

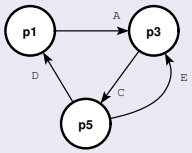
C.A Petri : Communication entre automates

Exemple introductif

2 processus (initialement) indépendants représentés par des automates.



A_1 est initialement en p1, il exécute cycliquement A, B puis C pour se réinitialiser.

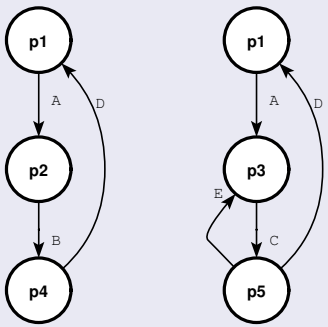


A_2 est initialement en p1, il exécute A puis C et arrive en p5. Il peut exécuter E et revenir en p3 ou choisir de faire D et de se réinitialiser.

Quizz :

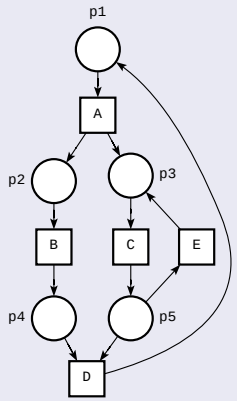
On veut contraindre A_1 et A_2 à commencer (action A) et à finir (action F) ensemble. Quel sera le comportement global ?

Réseaux de Petri



A_2 & A_2

$= Synchronisation \Rightarrow$



Réseau Place/Transition

Définition Réseau Place/Transition

$R =< P, T, Pre, Post >$ où

- P est un ensemble fini de *Places*
- T est un ensemble fini de *Transitions*
  $P \cap T = \emptyset$
- $Pre : P \times T \mapsto \mathbb{N}$ *incidence avant*
- $Post : P \times T \mapsto \mathbb{N}$ *incidence arrière*

Marquage / Réseau Marqué

Le réseau donne les règles de fonctionnement du système, le marquage donne l'état du système

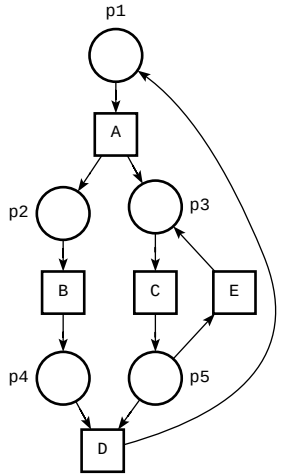
Marquage

$M : P \mapsto \mathbb{N}$ $\quad (M \in \mathbb{N}^P)$

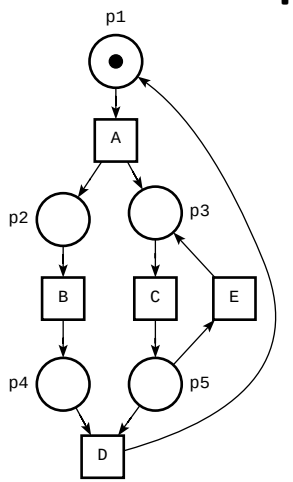
$M(p)$ dénote le marquage de la place p

Réseau marqué $N =< R, M >$

Graphiquement



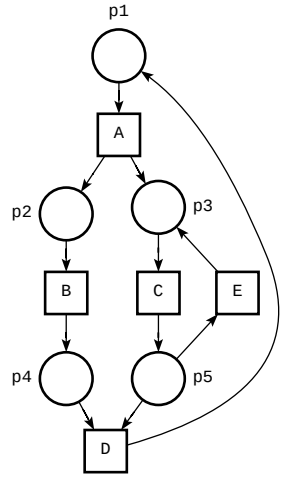
... avec le marquage



Définition Alternative

Système d'addition de vecteurs
$R =< P, T >$ où

- P est un ensemble fini de *Places*
- T est un ensemble fini de *Transitions*
  $T \subset \mathbb{N}^P \times \mathbb{N}^P$
  notations $t \in T = ({}^\bullet t, t^\bullet)$



Exemple

$T = \{A, B, C, D\}$

$$\begin{array}{ccccc} A & & B & & C \\ \left(\begin{vmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{vmatrix}, \begin{vmatrix} 0 \\ 1 \\ 1 \\ 0 \\ 0 \end{vmatrix}\right) & \left(\begin{vmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{vmatrix}, \begin{vmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 0 \end{vmatrix}\right) & \left(\begin{vmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \end{vmatrix}, \begin{vmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{vmatrix}\right) & \left(\begin{vmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 1 \end{vmatrix}, \begin{vmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{vmatrix}\right) & \left(\begin{vmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 1 \end{vmatrix}, \begin{vmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{vmatrix}\right) \end{array}$$

Remarques

Avec cette définition équivalente, les pre $({}^\bullet t)$ et post $(t^\bullet)$ conditions d'une transition ont le même type qu'un marquage : ${}^\bullet t, t^\bullet, M \in \mathbb{N}^P$

Sémantique des réseaux Place-Transition

Sémantique

La sémantique précise comment le modèle évolue :
- Quelles actions (transitions) sont possibles (sensibilisées) en fonction de l'état du système (marquage) ?
- Quel est l'effet d'une action (du tir d'une transition) sur l'état du système (marquage) ?

Sensibilisation d'une transition

Une transition $t({}^\bullet t, t^\bullet)$ est sensibilisée pour le marquage M ssi $M \geq {}^\bullet t$
On le note $M \xrightarrow{t}$

Tir d'une transition (franchissement)

Soit t une transition et M un marquage tel que $M[t>$
alors le tir de t à partir de M conduit en M' défini par : $M' = M -{}^\bullet t + t^\bullet$
On le note $M \xrightarrow{t} M'$

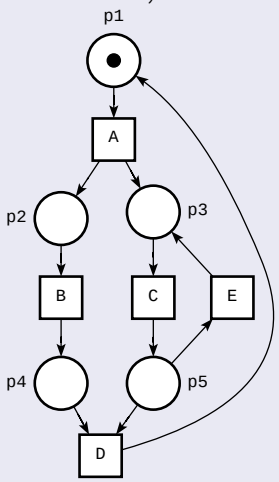
Remarques

+ ${}^\bullet t$ *est le marquage minimum permettant de franchir t*
+ $t^\bullet$ *est le marquage minimum atteint après avoir franchi t*

Exemples de sensibilisation/tir

Exemple de Sensibilisation

$< N, M_0 >$



Sensibilisation
$M_0 \geq^{\bullet} A$
donc $M_0 \xrightarrow{A}$

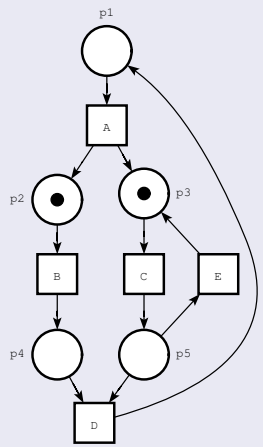
$$M_0 = \begin{vmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{vmatrix}, {}^{\bullet}A = \begin{vmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{vmatrix}$$

Exemple de Tir

Tir
$M_0 \xrightarrow{A} M_1$
$M_1 = M_0 -^{\bullet} A + A^{\bullet}$

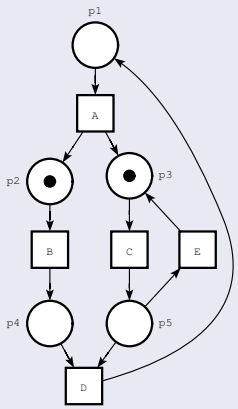
$$A^{\bullet} = \begin{vmatrix} 0 \\ 1 \\ 1 \\ 0 \\ 0 \end{vmatrix} \quad M_1 = \begin{vmatrix} 0 \\ 1 \\ 1 \\ 0 \\ 0 \end{vmatrix}$$

$< N, M_1 >$



Non déterminisme

$< N, M_1 >$



$M_1 \xrightarrow{B}$ **et** $M_1 \xrightarrow{C}$
$M_1 \geq^{\bullet} B$
$M_1 \geq^{\bullet} C$

M_1	${}^{\bullet}B$	${}^{\bullet}C$
0	0	0
1	1	0
1	0	1
0	0	0
0	0	0

Parallélisme (structurel/effectif)

Deux transitions t, t' sont *parallèles* ssi ${}^{\bullet}t \otimes^{\bullet} t' = \vec{0}$
où $(\vec{v} \otimes \vec{v'})(p) =_{Def} \vec{v}(p) \times \vec{v'}(p) \quad (\forall p \in P)$
Ici : B et C sont parallèles et pour M_1, le parallélisme est *effectif*

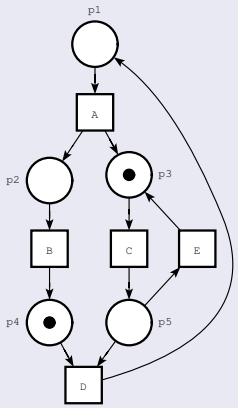
Conflict (dual du parallélisme)

Deux transitions t, t' sont *en conflict* ssi ${}^{\bullet}t \otimes^{\bullet} t' \neq \vec{0}$

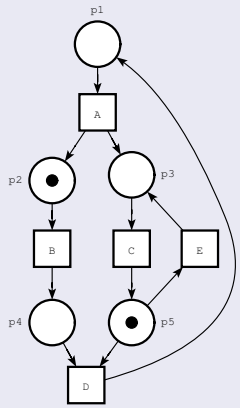
Exemple de Parallélisme

$M_1 \xrightarrow{B} M_2$
$M_1 \xrightarrow{C} M_3$
Comme $B//C$
On a $M_3 \xrightarrow{B}$ et $M_2 \xrightarrow{C}$
De plus
$M_3 \xrightarrow{C} M_4$ et $M_3 \xrightarrow{B} M_4$

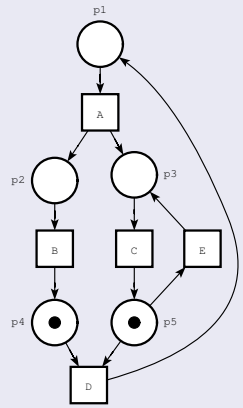
$< N, M_2 >$



$< N, M_3 >$

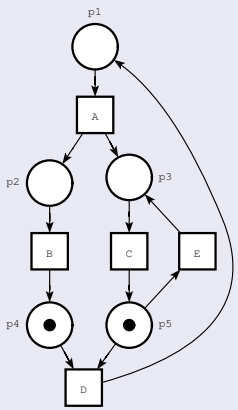


$< N, M_4 >$



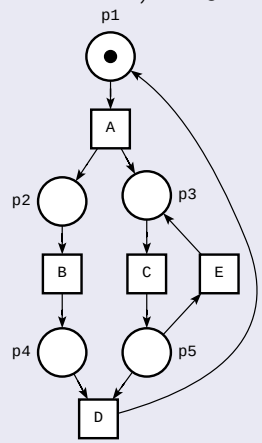
Exemples de Conflict

$< N, M_4 >$

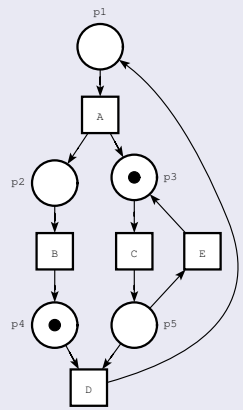


$M_4 \xrightarrow{D}$ et $M_4 \xrightarrow{E}$
mais $M_4 \not\xrightarrow{D.E}$
et $M_4 \not\xrightarrow{E.D}$
$M_4 \xrightarrow{D} M_0$ (et $M_0 \not\xrightarrow{E}$)
$M_4 \xrightarrow{E} M_2$ (et $M_2 \not\xrightarrow{D}$)
D et E sont en conflict en M_4

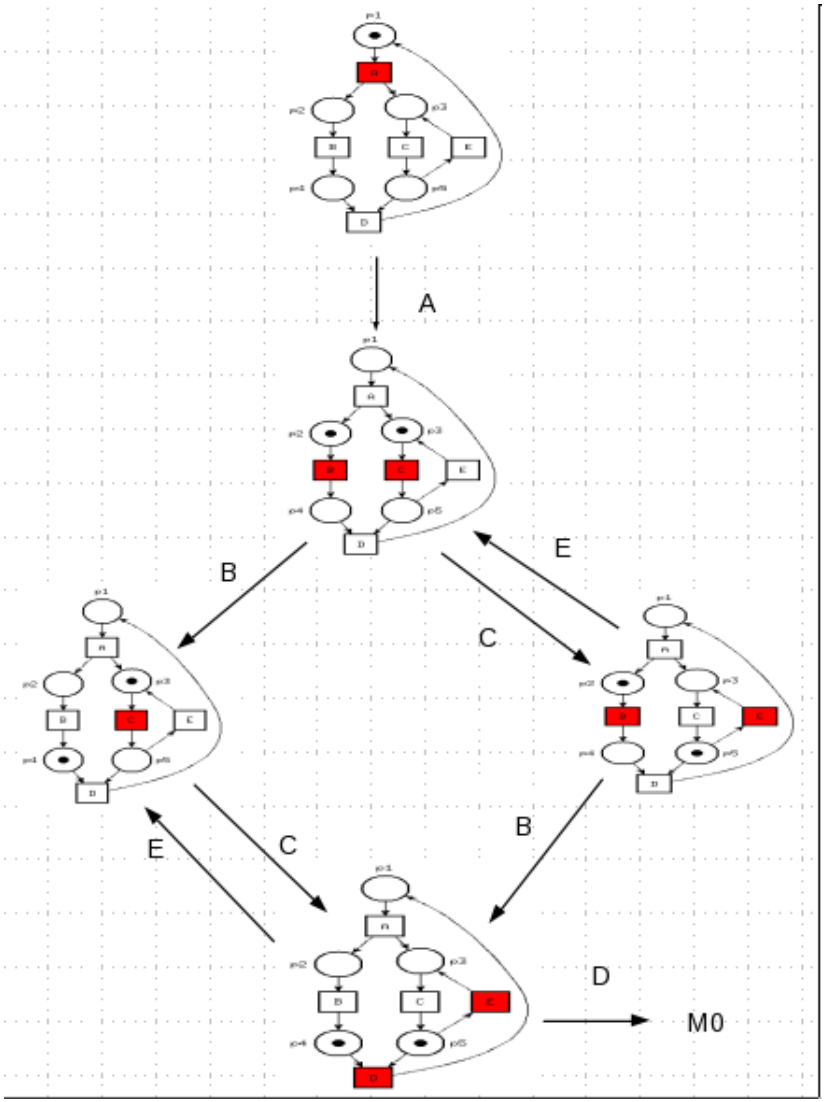
$< N, M_0 >$

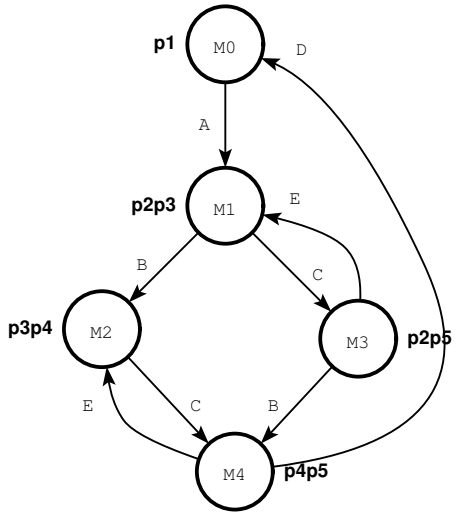


$< N, M_2 >$



Comportement d'un RdP (intuition)





M_0	M_1	M_2	M_3	M_4
1	0	0	0	0
0	1	0	1	0
0	1	1	0	0
0	0	1	0	1
0	0	0	1	1
p1	**p2p3**	**p3p4**	**p2p5**	**p4p5**

Graphe des Marquages accessibles

Ensemble des Etats Accessibles : $A(R, m_0)$

$A_0 = \{m_0\}$
$A_i = \{m \in \mathbb{N}^P / \exists p \in A_{i-1}, \exists t \in T : p[t > m\}$
$A(R, m_0) = \bigcup_{i \geq 0} A_i$

Rien n'assure la convergence de la suite ! $A(R, m_0)$ **peut être infini**

Graphe des Etats Accessibles : $G(R, m_0)$

$G(R, m_0) =< A(R, m_0), \rightarrow, L >$ où

- $A(R, m_0)$: ensemble des sommets
- $\rightarrow$: la relation de transition est le plus petit ensemble vérifiant :
  $(m_1, t, m_2) \in \rightarrow$ ssi $m_1, m_2 \in A(R, m_0), t \in T$ et $m_1[t > m_2$
- L : ensemble des labels de transition :
  $L \subset T$ défini par $t \in L$ ssi $m_1[t > m_2$

Langage associé à un réseau $L(R, m_0)$

$L(R, m_0) = \{\sigma \in T^* : m_0 \xrightarrow{\sigma}\}$

Semi-algortihme de calcul de $G(R, m_0)$

semi-algorithme : car rien n'asssure la terminaison

Semi-Algorithme

– 1. Initialisation :

$Stack \leftarrow \emptyset$; push m_0 into $Stack$

$A(R, m_0) \leftarrow \emptyset$; enter m_0 in $A(R, m_0)$

– 2. Exploration :

while $Stack \neq \emptyset$

loop {

pop(q) from stack

$TS \leftarrow \{t \in T : q\ [\ t\ >\}$;

$\forall t \in\ TS$ *do*

$\{q\ [\ t\ > q'$;

if $q' \notin A(R, m_0)$

then $\{$*enter* q' *in* $A(R, m_0)$; *put* q' *onto Stack*$\}$

enter (q, t, q')*in* $\rightarrow$

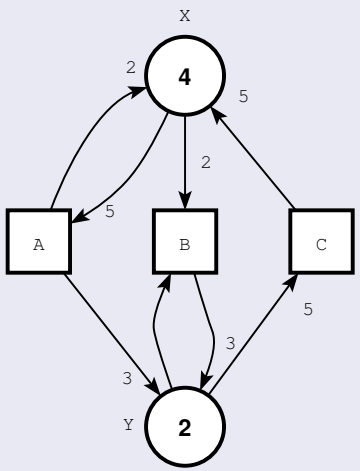
$\}$

}

Exercice : Calcul de $G(N, M_0)$

nb : les arcs de N sont valués

$< N, n_0 >$



Q1 : Complétez la description des transitions $(^\bullet t, t^\bullet)$

m_0

$\begin{vmatrix} 4 \\ 2 \end{vmatrix}$

X^4Y^2

	A	B	C
	$\left(\begin{vmatrix} 5 \\ 0 \end{vmatrix}, \begin{vmatrix} 2 \\ 3 \end{vmatrix}\right)$	$\left(\begin{vmatrix} 2 \\ 1 \end{vmatrix}, \begin{vmatrix} \\ \end{vmatrix}\right)$	$\left(\begin{vmatrix} \\ \end{vmatrix}, \begin{vmatrix} \\ \end{vmatrix}\right)$

Q2 : Exploration de X^4Y^2

$X^4Y^2 \xrightarrow{A} \not\rightarrow$

$X^4Y^2 \xrightarrow{B} X^2Y^4$

$X^4Y^2 \xrightarrow{C} ?$

Q3 : Exploration de X^2Y^4

$X^2Y^4 \xrightarrow{A} ?$

$X^2Y^4 \xrightarrow{C} ?$

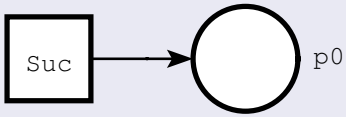
$X^2Y^4 \xrightarrow{B} ?$

Suite

Q4 : Explorez X^0Y^6
Q5 : Explorez X^5Y^1
Q5 : Explorez X^3Y^3
Q6 : Finissez la construction du graphe

Réseaux bornés

Quizz : Soit le réseau $< \mathbb{N}, [0] >$



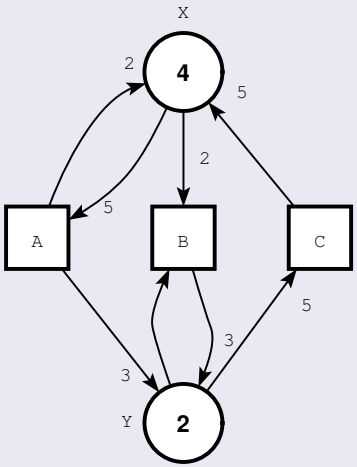
Q1 : A-t-on $[0] \xrightarrow{Suc} ?$
Q2 : l'ensemble des marquages de ce réseau est-il fini ?

Réseau borné

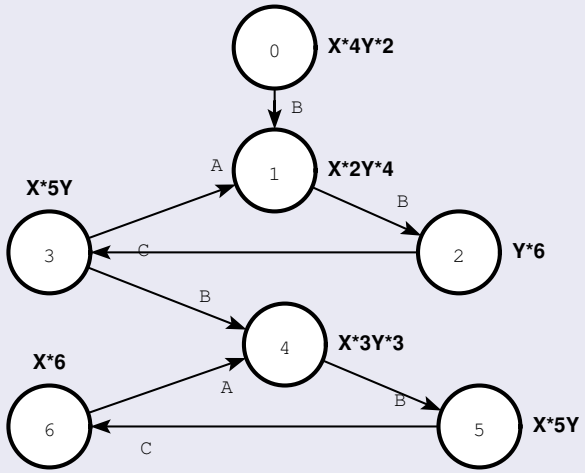
Un réseau marqué $< N, M_0 >$ est borné ssi
$\exists b \in \mathbb{N} \ : \ \forall p \in P, \forall M \in A(N, M_0) : M(p) \leq b$

Un exemple de réseau 6-borné

$< N, n_0 >$



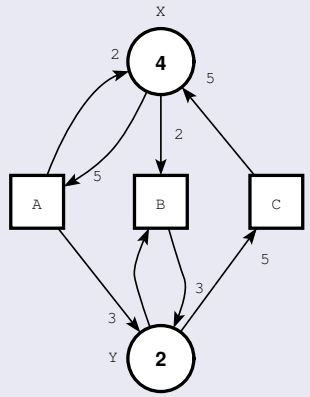
$G < N, n_0 >$



Réseau structurellement borné

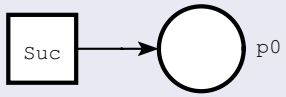
Un réseau N structurellement borné ssi $\forall m_0 \in \mathbb{N}^P < N, m_0 >$ est borné

Exemple de réseau structurellement borné



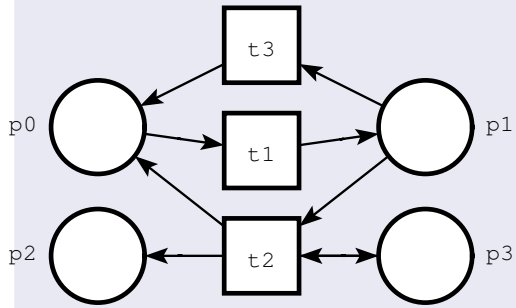
Le nombre global de jetons dans le réseau est invariant quel que soit le marquage ; chaque transition modifie uniquement la localisation des jetons dans les places mais laisse leur nombre inchangé.

Plus petit réseau **non** borné : $< \mathbb{N}, [0] >$



Il permet de générer $\mathbb{N}$
Il est même structurellement **non** borné

En général, la propriété bornée dépend du marquage initial (cf réseau G)



Trouvez la condition la plus générale sur m_0 pour que :

- $< G, m_0 >$ soit borné
- $< G, m_0 >$ soit non borné

Réseaux bornés et Finitude de l'espace d'états

Propriété : $G(R, m_0)$ est fini ssi $< R, m_0 >$ est borné

($\Rightarrow$) Soit $b = Max(m)$ pour $m \in A(R, m_0)$
où $Max(m) = max(m(p))$ pour $p \in P$
$< R, m_0 >$ est b-borné

($\Leftarrow$) Soit b la borne de $< R, m_0 >$
alors $A(R, m_0) \subset [0, b]^P$ qui contient $(b+1)^{|P|}$ éléments
$A(R, m_0)$ et $G(R, m_0)$ sont donc finis

Propriété

La finitude de l'espace des états accessibles - et ce qui revient au même - la propriété k-borné sont décidables
nb : Très important pour adresser des problèmes de "dimensionnement"

$\mapsto$ Arbre/Graphe de couverture [Karp& Miller 1969]
Construction "vue" plus tard

Outline

Piscine

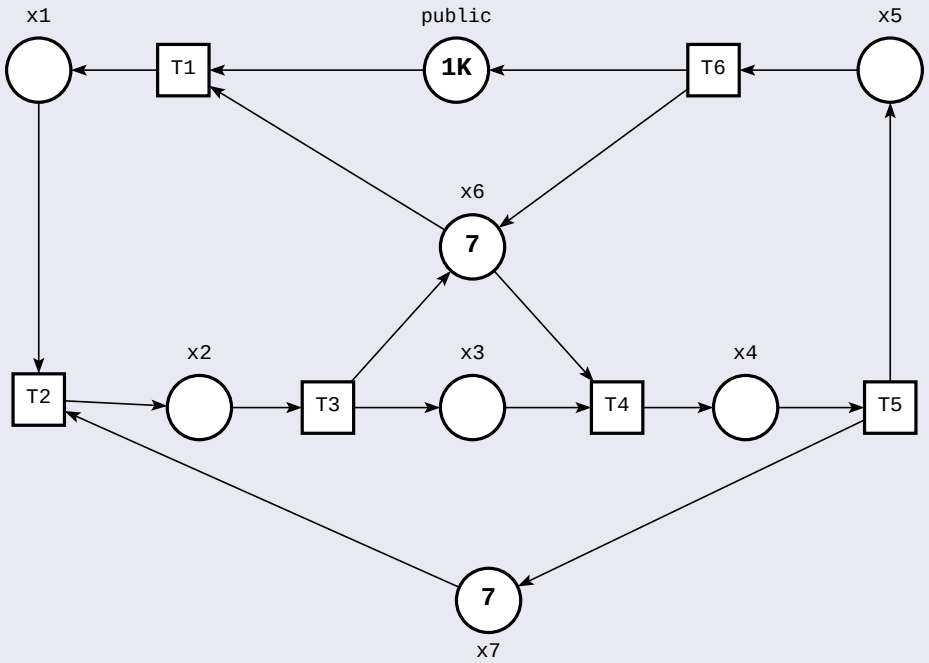
Enoncé

Une piscine comporte c cabines pour se changer et p paniers pour déposer ses vétements.

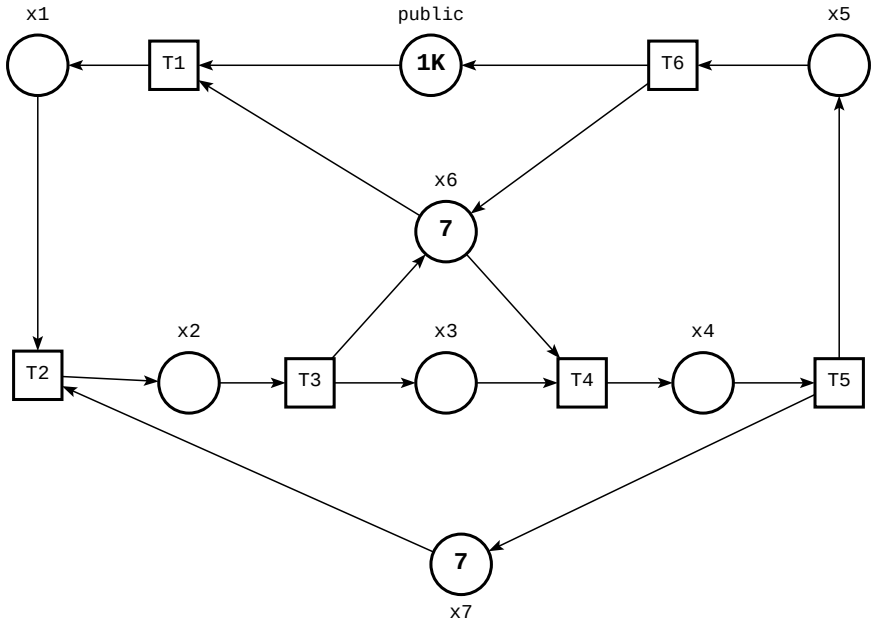
- On n'entre dans le piscine que si une cabine est libre. On attend un panier pour se changer et déposer ses vètements. On libère la cabine et on pénètre dans le bassin.

- On ne quitte le bassin que si une cabine est libre. On se change et on restitue cabine et panier. On quitte la piscine.

Un modèle possible



Piscine : Légende du réseau



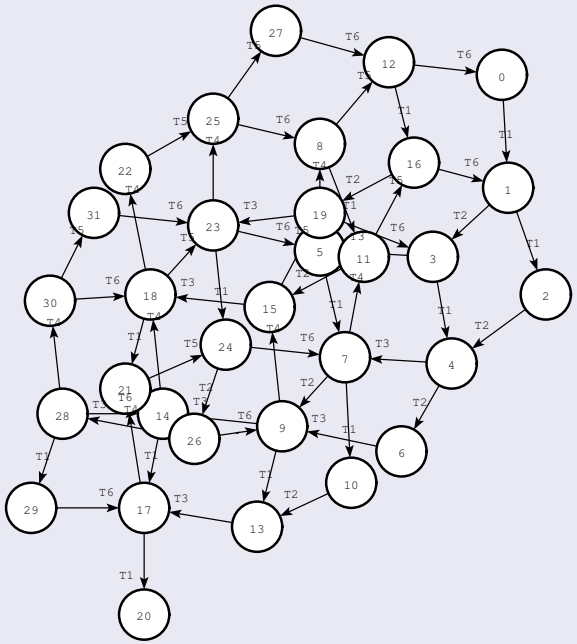
T1 : le client entre à la piscine
(si ∃ cabine libre (x6))
T2 : le client se déshabille
(si ∃ panier libre (x7))
T3 : le client pénètre dans le bassin
(et restitue la cabine (x6))
T4 : le client quitte le bassin
(si ∃ cabine libre (x6))
T5 : le client s'habille
(et restitue le panier (x7))
T6 : le client quitte la piscine.

x1 : client en attente d'un panier
x2 : client se déshabille
x3 : client dans le bassin
x4 : client s'habille
x5 : client habillé prêt à sortir
x6 : Compteur de cabines libres
x7 : Compteur de paniers libres

Piscine : Le système peut-il se bloquer ?

Graphe des marquages pour $c = p = 2$

Graphe : 32 marquages, 57 transitions



1 seul état de Blocage : x1(2) x3(2)

Explosion combinatoire

Pour $n = k = 10 \mapsto$ 7006 marquages, 28885 transitions
Pour $n = k = 15 \mapsto$ 38759 marquages 178703 transitions
et toujours un seul état de blocage ....

Une histoire de pont

Enoncé

Un groupe de 4 personnes (A,B,C,D) se situe sur la rive gauche d'un fleuve et doit se rendre sur la rive droite.
Pour ce faire, ils doivent emprunter un pont mal éclairé qui ne peut supporter qu'une charge de deux personnes. Le groupe dispose d'une seule lampe de poche. Chaque traversée du pont nécessite la possession de la lampe. Il est donc nécessaire de ramener la lampe de l'autre coté pour permettre une nouvelle traversée.
Les 4 personnes marchent à une vitesse différente. Les temps de traversée pour chacun des individus est respectivement de 10 min pour A, 5 min pour B, 2 min pour C et 1 min pour D.

Question : Quel est le temps minimal pour faire passer les 4 personnes sur la rive droite ?

Principe de la modélisation

Les Places

- *"booléens"* : Lg, Ag, Bg, Cg, Dg, Lg, Ag, Bg, Cg, Dg
indiquant la localisation (gauche/droite) des personnes et de la lampe.
Initialement : Lg Ag Bg Cg Dg
- *"compteur"* : Temps permet de cumuler le temps écoulé depuis le début
Initialement : Temps*0

Les Transitions

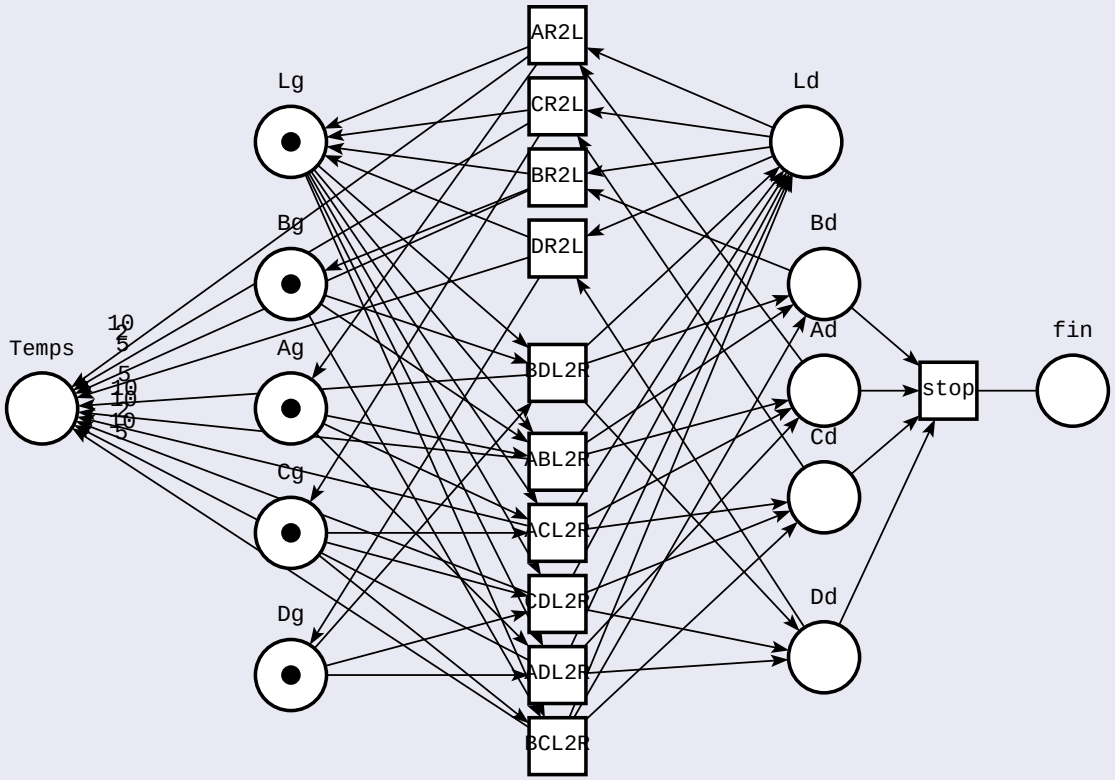
6 transitions de la forme XYL2R représentant le passage de gauche à droite (L2R) des personnes X et Y. Il faut que la lampe ainsi que X et Y soient à gauche, ils sont à droite après le tir. Le temps de la traversée (le max des durées de passage) est ajouté 'à Temps.

4 transitions de la forme YR2L représentant le retour (R2L). Il faut que la lampe et Y soint à droite. Ils sont à gauche après le tir. Le temps de traversée de Y est ajouté à Temps.

1 transition stop matérialisant la fin du "calcul"

Modélisation

Un modèle (perfectible) possible



Propriétés attendues

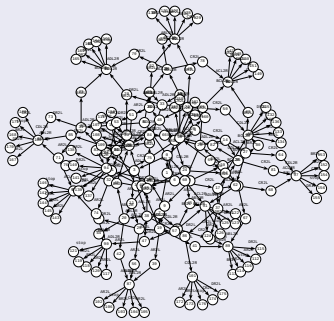
Ici on a un modèle de calcul (!)

Terminaision : atteindre inévitablement un deadlock

Correction partielle : tous les états de deadlock doivent marquer la place Fin

Résultat de l'analyse

Le comportement (graphe des marquages)



Taille : 177 marquages, 237 transitions

Borné, Bloquant (et c'est tant mieux !)

Le système se bloque inévitablement (terminaison)

Le système peut être bloqué sans marquer la place Fin (pas de correction partielle)

Temps minimal de Traversée : 17

Une analyse des "états terminaux" permet de récupérer les $\neq$ temps de "transit"

Temps d'exécution possibles : **17**, 19, 20 , 21, 23, 24, 26, 27, 30, 33, 34, 36, 37, 40, 50

Traversée en 17 u.t

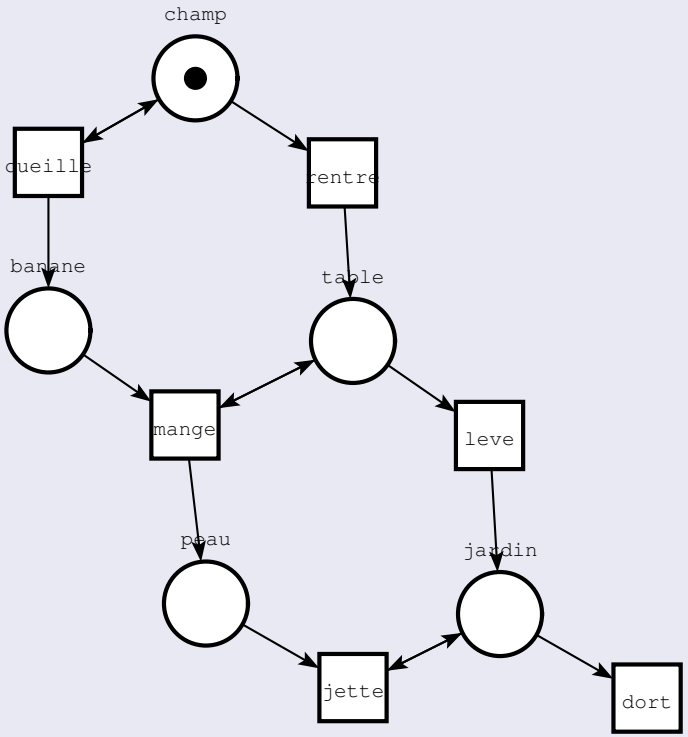
Passage de C et D	2 min
Retour de C	4 min
Passage de A et B	14 min
Retour de D	15 min
Passage de C et D	17 min

Exemple de Réseau non borné

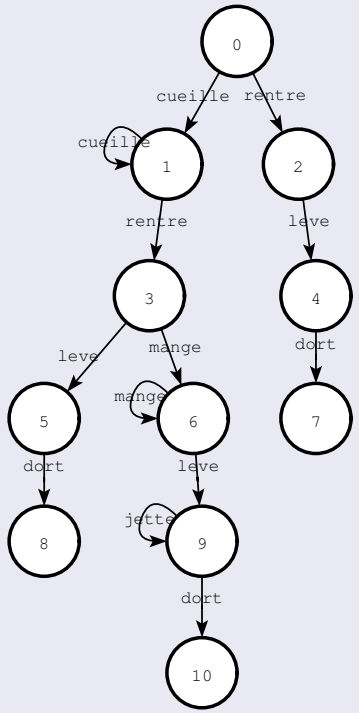
Journée d'un planteur de bananes

Le planteur est initialement au champ où il cueille des bananes (en quantité supposé infinie). Il cesse son travail et se met à table pour manger quelques bananes. A la fin du repas, il passe au jardin et jette quelques unes des peaux de bananes mangées. Il part ensuite dormir.

(Un) réseau possible



Son arbre de couverture



AC : Sur-approximation du

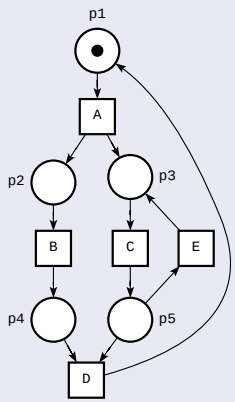
Outline

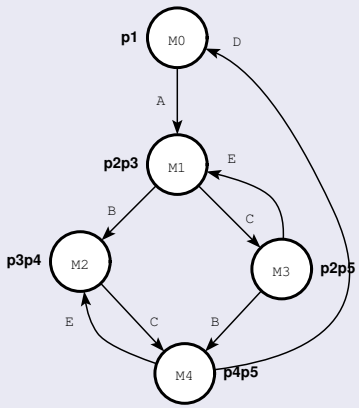
Propriétés générales d'accessibilité

Blocage / Deadlock / Puits

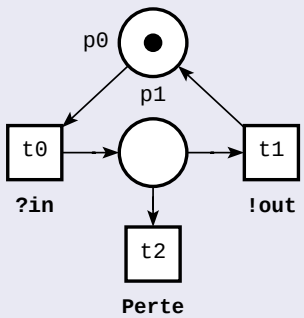
N est sans blocage ssi $\forall m \in G(R, m_0), \exists t \in T : m[\ t\ >$

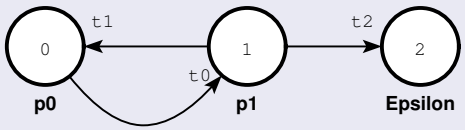
$N\#1$: Exemple de réseau **non bloquant**





$N\#2$: Exemple de réseau **bloquant**



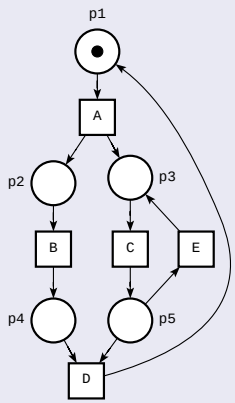


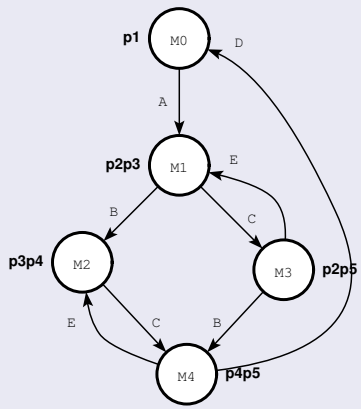
Réinitialisable (propre)

N est Réinitialisable (propre) ssi $\forall m \in G(R, m_0), \exists \omega \in T^+ : m \xrightarrow{\omega} m_0$

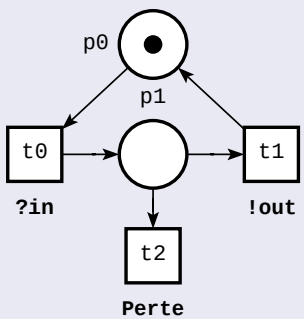
nb : $\omega \in T^+$ impose que ω soit non nulle

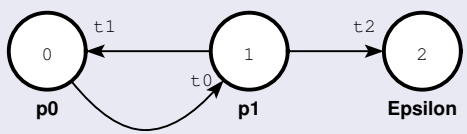
$N\#1$: réseau **propre**





$N\#2$: réseau **non propre**



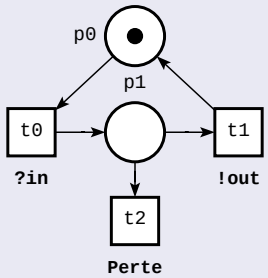


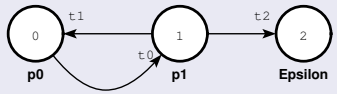
Quasi-vivacité

1. Une transition t est quasi-vivante ssi $\exists m \in G(R, m_0) : m \xrightarrow{t}$
2. Le réseau est quasi-vivant si chacune de ses transitions est quasi-vivante.

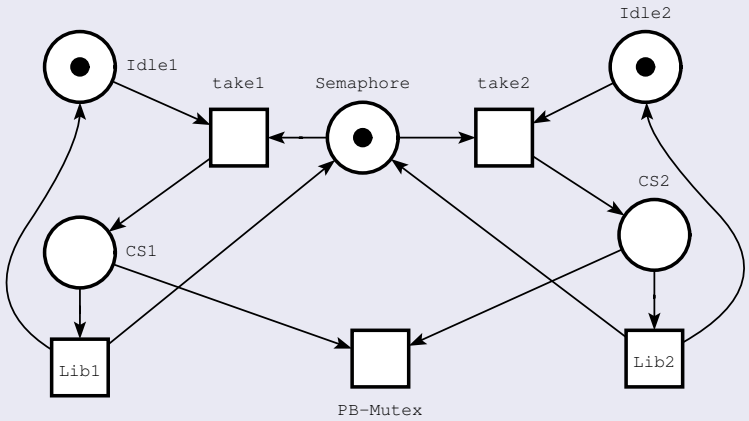
Un transition non vivante est dite **morte**

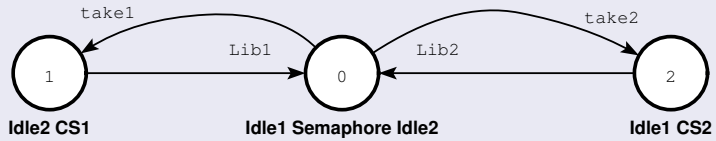
$N\#2$: réseau **quasi-vivant**





$N\#3$: Réseau **non quasi-vivant** : transition PB-mutex **morte**





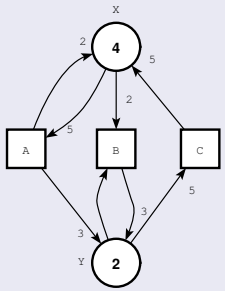
nb : le fait que pb-mutex soit morte permet de vérifier la propriété d'exclusion mutuelle (principe de vérification par la méthode des observateurs)

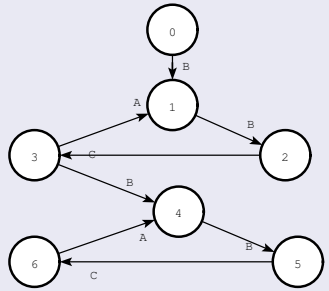
N est Vivant

1. Une transition t est **vivante** ssi $\forall m \in G(R, m_0) : \exists \omega \in T^* : m \xrightarrow{\omega.t}$
2. Le réseau est vivant si chacune de ses transitions est vivante.

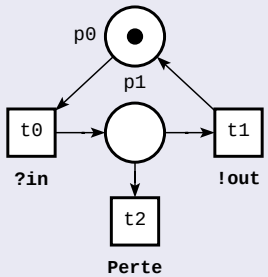
Un transition non vivante est dite **non vivante** (!)

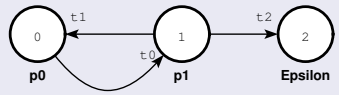
$N\#4$: réseau non réinitialisable mais **vivant**



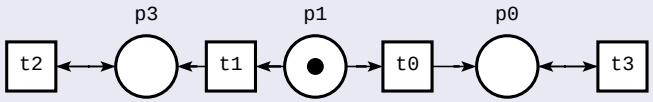


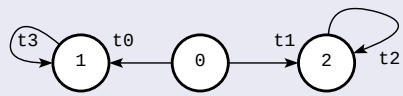
$N\#2$: Réseau **quasi-vivant**, bloquant et non vivant





$N\#5$: **quasi-vivant**,**non bloquant**, aucune transition vivante



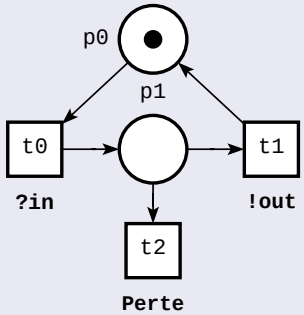


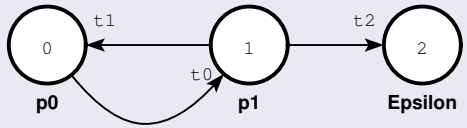
Infiniment actif

Un réseau est infiniment actif ssi $\forall k \in \mathbb{N}, \exists \sigma \in T^K : m \xrightarrow{\sigma}$

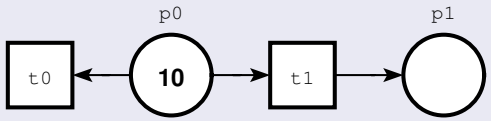
Le réseau admet au moins une séquence de franchissement infinie.

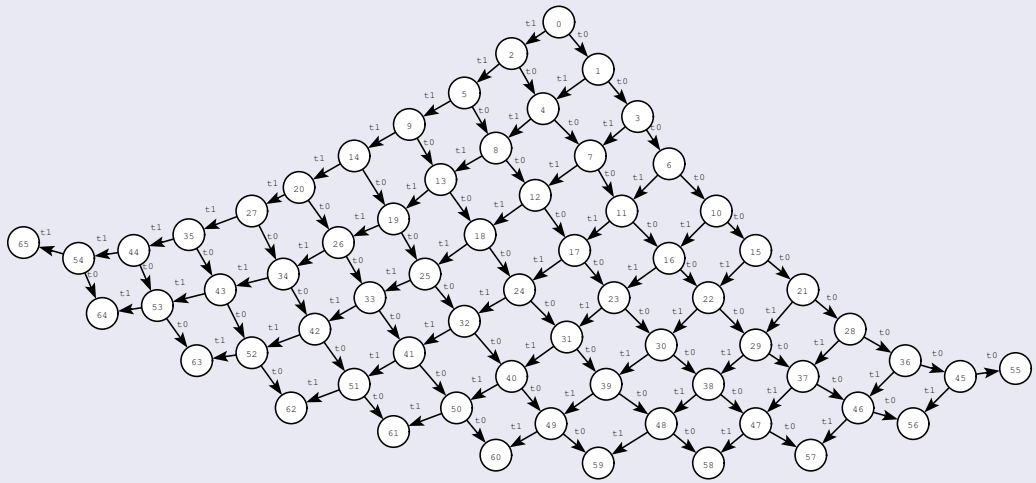
$N\#2$: Réseau **bloquant** et **infiniment actif**





$N\#6$: Réseau **non infiniment actif**



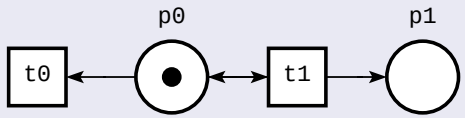


Toutes les séquences sont de longueur bornées (ici par 10)

Lien entre ces différentes propriétés

- Réinitialisable $\Rightarrow$ Sans blocage
- Vivant $\Rightarrow$ Sans blocage
- Réinitialisable et Vivant **sont sans rapport**
  $N\#4$ est vivant sans être réinitialisable
  $N\#3$ est réinitialisable sans être vivant
- Réinitialisable + Quasi-Vivant $\Rightarrow$ Vivant
- Sans blocage $\Rightarrow$ Infiniment actif
  Réciproque fausse (cf $N\#2$)
- Pas de rapport entre Infiniment actif et quasi-vivacité, vivacité, propre
- Pas de rapport entre borné et bloquant, quasi-vivacité, vivacité, propre
  Réseau $N\#7$ est non borné et contient une infinité d'états de blocage
- Non borné $\Rightarrow$ Infiniment actif
  Réciproque fausse (par ex $N\#2$)

Réseau $N\#7$: non borné et avec une infinité de deadlocks



$$A(N\#7, p0.p1) = \{p0.p1^k : k \in \mathbb{N}\} \cup \{p1^k : k \in \mathbb{N}\}$$

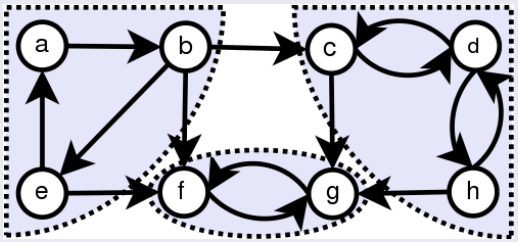
$\{p1^k : k \in \mathbb{N}\}$ contient les deadlocks

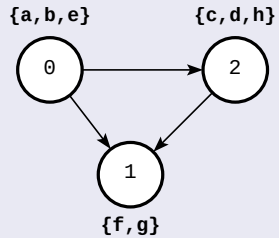
Décision des propriétés générales via les CFC

(Rappels) Composantes fortement connexes (CFC/SCC)

Etant donné un graphe orienté $G =< S, \rightarrow >$

1. $\mathcal{C} \subset S$ est une composante fortement connexe de G ssi $\mathcal{C}$ est un sous-ensemble **maximal** vérifiant : $u, v \in \mathcal{C}$ alors $\exists \sigma_1, \sigma_2 \in T^* : u \xrightarrow{\sigma_1} v$ et $v \xrightarrow{\sigma_2} u$
2. Les CFC forment une partition du graphe et induisent une relation d'équivalence $\sim_{cfc}$: $s \sim_{cfc} q$ ssi s et q appartiennent à la même CFC.
   Sur l' exemple $S/\sim_{cfc} = \{\{a, b, e\}, \{c, d, h\}, \{f, g\}\}$
3. On peut " quotienter" le graphe initial ($G =< S, \rightarrow >$) pour obtenir le graphe quotient des CFC : $G/\sim_{cfc} =< S_{\sim_{cfc}}, \rightarrow_{\sim_{cfc}} >$ défini par :
   $S_{\sim_{cfc}} = S/\sim_{cfc}$ et $\rightarrow_{\sim_{cfc}} = \{(s, q) \in \rightarrow : s \not\sim_{cfc} q\}$





4. Une cfc est "pendante" ssi $\forall e \in cfc, e \rightarrow q \Rightarrow q \in cfc$
   $\{f, g\}$ est pendante
5. Une cfc est "triviale" ssi elle est réduite à un point et ne contient pas de boucle.

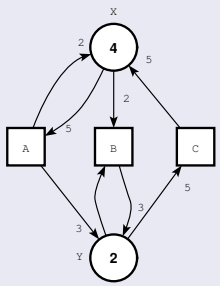
Décision de (certaines) propriétés générales via les CFC

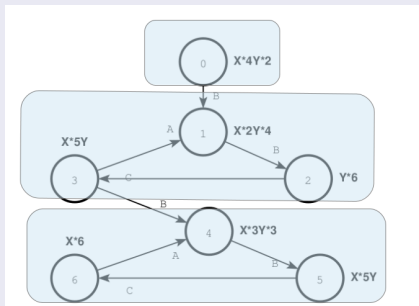
Soit $< R, m_0 >$ un réseau borné :

1. $< R, m_0 >$ est bloquant ssi $G(R, m_0)/\sim_{cfc}$ contient au moins une cfc pendante triviale.
2. $< R, m_0 >$ est propre ssi $G(R, m_0)/\sim_{cfc}$ contient une seule cfc.
3. $< R, m_0 >$ est infiniment actif ssi $G(R, m_0)/\sim_{cfc}$ contient au moins une cfc non triviale.
4. $< R, m_0 >$ est vivant ssi chaque cfc pendante de $G(R, m_0)/\sim_{cfc}$ contient toutes les transitions.

nb : les cfc n'apportent pas de réponse sur la quasi-vivacité ou sur le caractère borné

Retour sur $N\#4$

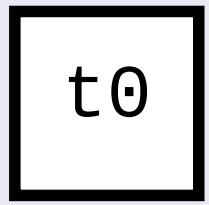




- 3 cfc donc $N\#4$ n'est pas réinitialisable
- chaque cfc pendante (il n' y en a qu'une) contient toutes les transitions : $N\#4$ est vivant
- 2 cfc non triviales donc $N\#4$ est infiniment actif

Réseau $N\#0$

Ceci est un réseau constitué par une unique transition (pas de place)



Exercice : Complétez le tableau suivant

N	#0	#1	#2	#3	#4	$N\#5$	#6	#7
Bloquant								
Réinitialisable								
Quasi-Vivant								
Vivant								
Infiniment Actif								
Borné								

Outline

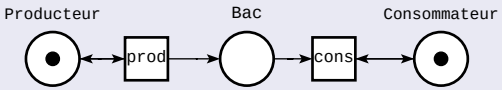
Quelques rappels ....

Réseau borné

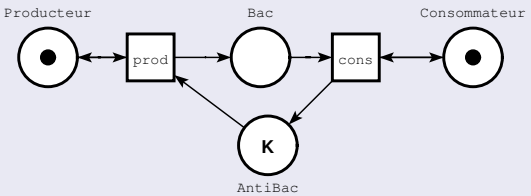
Soit $< N, M_0 >$ un réseau marqué

1. Une place $p \in P$ est b-bornée ssi ssi $\exists b \in \mathbb{N}, \forall M \in A(N, M_0) : M(p) \leq b$
2. $< N, M_0 >$ est b-borné ssi chacune de ses places est b-bornée
3. $< N, M_0 >$ est borné ssi $\exists b \in \mathbb{N} : < N, M_0 >$ est b-borné

$N\#8$: Producteur/Consommateur



Place Bac est non bornée



Les places Bac et AntiBac sont complémentaires et sont k-bornées

Remarques

Le semi-algorithme de construction du graphe des marquages n'apporte pas de réponse.
La propriété "borné " est importante puisqu'elle conditionne la possibilité d'implantation du modèle

Décision de la propriété k-borné

Les grandes étapes

- Propriétés de Monotonie
- Caractérisation des réseaux infiniment Actifs
- Algorithme de Karp & Miller
  Arbre de Couverture & Graphe de Couverture

Monotonie : Soit un réseau marqué $< N, m_0 >$

Sensibilisation/Tir (rappels)

1. $t({}^\bullet t, t^\bullet)$ est sensibilisée pour le marquage M ssi $M \geq^\bullet t$
2. Si $M \geq^\bullet t$ alors $M \xrightarrow{t} M'$ avec $M' = M -^\bullet t + t^\bullet$

nb : Si $X \geq M$ alors $X \xrightarrow{t} X'$ et $X' - X = M' - M = -^\bullet t + t^\bullet$

$A(N, m_0), G(N, m_0), L(N, m_0)$

Soit $M \geq m_0$ alors
$|A(N, m_0)| \leq |A(N, M)|$, $|G(N, m_0)| \leq |G(N, M)|$, $L(N, m_0) \subseteq L(N, M)$

Caractérisation des réseaux Infiniment actifs

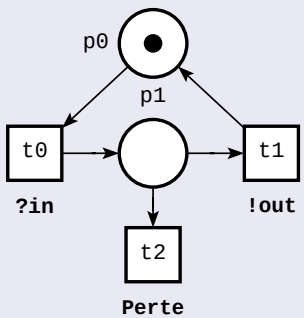
Séquences répétitive

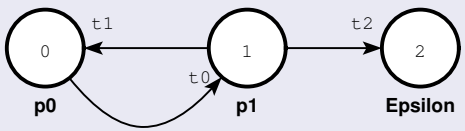
Une séquence de transitions $\sigma \in T^*$ est dite répetitive ssi

$$\forall M \in A(N, m_0) : M \xrightarrow{\sigma} M' \Rightarrow M' \xrightarrow{\sigma}$$

Prop : $\sigma \in T^*$ est répetitive ssi $\forall M \in A(N, m_0) : M \xrightarrow{\sigma} M' \Rightarrow M' \geq M$

$N\#2$: bloquant, borné et infiniment actif





La séquence $t0.t1$ est répétitive ($X \xrightarrow{t0.t1} X \quad (\forall X)$) et franchissable

Caractérisation des réseaux Infiniment actifs

(rappel) Un réseau est infiniment actif ssi il admet au moins une séquence de franchissement infinie.

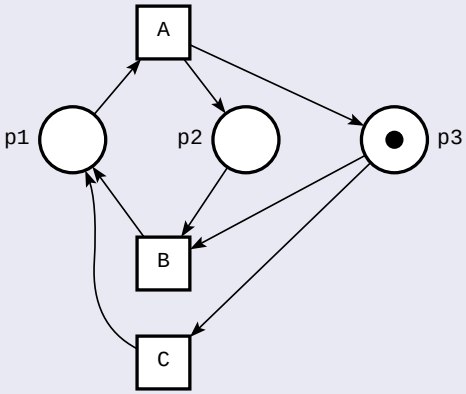
Prop : Un réseau est infiniment actif ssi il admet une séquence de franchissement répetitive

Séquence répétitive **croissante**

Séquence répétitive croissante pour une place donnée

Une séquence de transitions $\sigma \in T^*$ est dite répétitive croissante pour une place p ssi $\forall M \in A(N, m_0) : M \xrightarrow{\sigma} M' \Rightarrow M' \geq M$ et $M'(p) > M(p)$

$N\#9$: Un autre réseau non borné

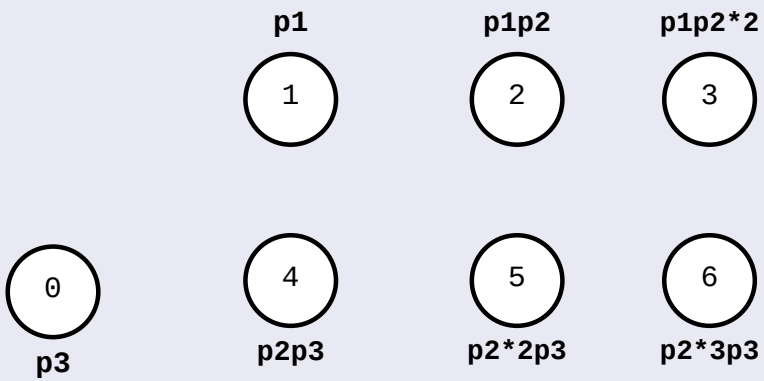


La séquence $C.A$ est répétitive croissante pour la place $P2$

$$\left|\begin{matrix}0\\0\\1\end{matrix}\right| \xrightarrow{C} \left|\begin{matrix}1\\0\\0\end{matrix}\right| \xrightarrow{A} \left|\begin{matrix}0\\1\\1\end{matrix}\right| \text{ avec } \left|\begin{matrix}0\\0\\1\end{matrix}\right| \leq \left|\begin{matrix}0\\1\\1\end{matrix}\right|$$

ou $p3 \xrightarrow{C} p1 \xrightarrow{A} p2p3$

Exercice : Exploration Partielle de $G(N\#9, p3)$



Reliez les marquages entre-eux

Donnez les(des) séquences répétitives croissantes

Caractérisation des réseaux non bornés

Caractérisation des réseaux non bornés

Une place p est non bornée ssi le réseau admet une séquence de franchissement répetitive croissante pour la place p

(rappel) Un réseau est non borné ssi il admet une place non bornée

Arbre de Couverture (Karp & Miller)

Arbre de Couverture : Structure finie permettant de détecter toutes les places non bornées et de donner une borne à toutes les autres.

Principe : Explorer l'espace d'états de façon à détecter toutes les séquences répétitives croissantes

1. Chaque marquage exploré sera comparé à tous ses prédécesseurs : on doit donc être capable de remonter dans le passé. On utilise donc un arbre au lieu d'un graphe
2. On veut être aussi capable de donner une borne pour les places bornées : on considère des pseudo-marquages ($\in (\mathbb{N} \cup \{\omega\})^P$) pour lesquels les places non bornées sont marquées par ω. On obtient ainsi une "couverture" (finie) de l'ensemble (infini) des marquages réels.

Arbre de couverture

Définition de $\mathbb{N}_\omega$ et des opérations afférentes

$\mathbb{N}_\omega = \mathbb{N} \cup \{\omega\}$
$\forall n \in \mathbb{N} : n < \omega$

$n + \omega = \omega + n = \omega$
$\omega - n = \omega$ ($n - \omega$ n'est pas défini)

On étend de la même manière $+, -, <$ à $\mathbb{N}_\omega^P$

Intuition de l'arbre de couverture ($AC(R, m_0)$)

Les sommets de l'arbre de couverture sont des ω-marquages ($\in {\mathbb{N}_\omega}^P$)
Propriété de couverture : $AC(R, m_0)$ "couvre" $A(R, m_0)$
$\forall m \in A(R, m_0), \exists m' \in AC(R, m_0) : m \leq m'$

Algorithme de Karp & Miller

Algorithme construction de $AC(R, m_0)$

Soit q un sommet étiqueté par m

Si q admet un ancètre p ayant la même étiquette

Alors q n'a pas de fils.

Sinon

Pour chaque transition t sensibilisée en m : $m \xrightarrow{t} m_t$

q admet un fils q_t,

l'arc reliant q à q_t est étiqueté par t

et le sommet q_t est étiqueté par $\Omega(m_t)$

Définition de $\Omega(m_t)(p)$

Pour chaque place $p \in P$

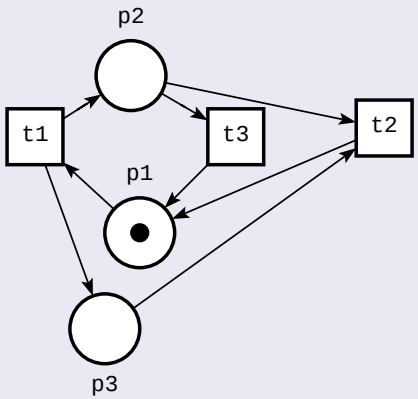
Si q_t admet un ancètre a étiqueté par m_a

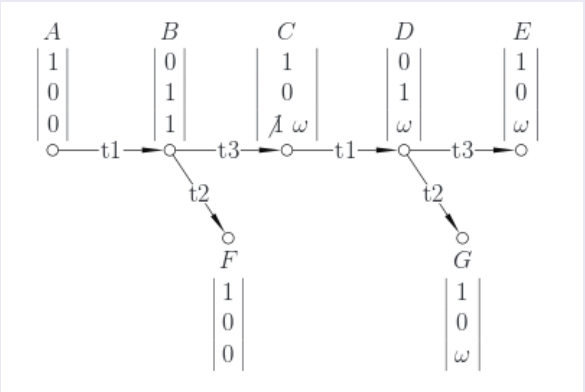
avec $m_a \leq m_t$ et $m_a(p) < m_t(p)$

Alors $\Omega(m_t)(p) = \omega$ Sinon $\Omega(m_t)(p) = m_t(p)$

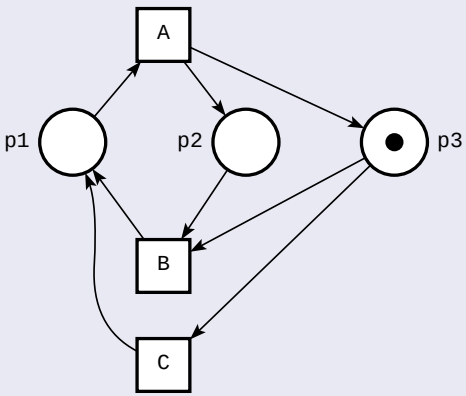
Exemples d'arbres de couverture

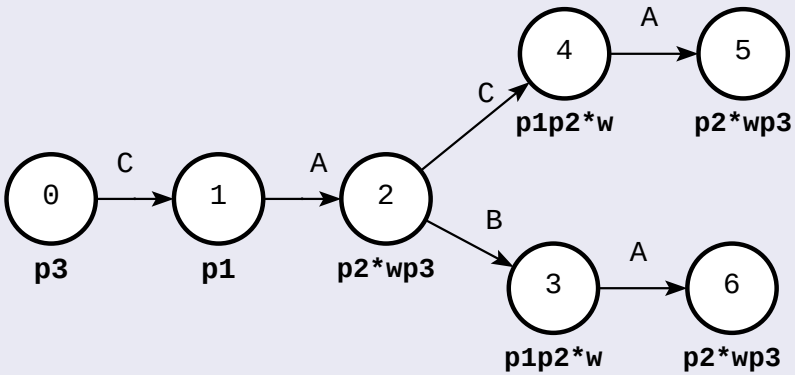
$N\#10$: Réseau non borné et son arborescence de couverture





$N\#9$ (déjà vu)





Arbre et Graphe de couverture

Propriétés de l'arbre de couverture [Karp-Miller 69]

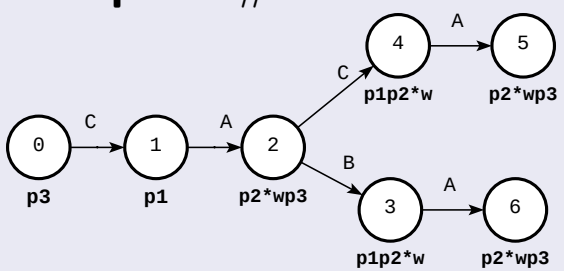
Soit un réseau marqué $< R, m_0 >$

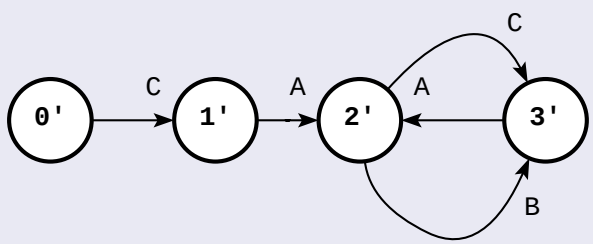
1. $AC(R, m_0)$ est fini
2. $< R, m_0 >$ est **non borné** ssi $\exists\ q \in AC(R, m_0), \exists p \in P$ tel que $q(p) = \omega$
3. Une place p est **non bornée** ssi $\exists q \in AC(R, m_0)$ tel que $q(p) = \omega$
4. Une place bornée p a pour borne $Max(\{q(p) : q \in AC(R, m_0)\})$

Graphe de Couverture $GC(R, m_0)$

$GC(R, m_0)$: Quotient de l'arbre de couverture obtenu en identifiant les sommets étiquetés par le même marquage

Exemple $N\#9$: Arbre et Graphe de couverture





$S/\sim = \{\{0\}, \{1\}, \{2, 5, 6\}, \{3, 4\}\}$

$2' \leftrightarrow \{2, 5, 6\}$ et $3' \leftrightarrow \{2, 4\}$

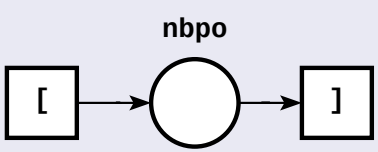
Graphe de Couverture

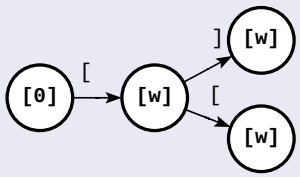
Propriétés du Graphe de Couverture $GC(R, m_0)$

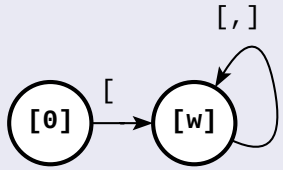
1. Si $< R, m_0 >$ est borné alors $GC(R, m_0) = G(R, m_0)$
2. $\forall \sigma \in T^*$ Si $m_0 \xrightarrow{\sigma} m$ dans $< R, m_0 >$
   alors dans $GC(R, m_0)$ on a aussi $m_0 \xrightarrow{\sigma} m$
   Réciproque fausse : Une séquence dans le graphe de couverture ne correspond pas forcément à une séquence de franchissement du réseau.

   L'introduction des ω-marquages peut se traduire par l'apparition de séquences de franchissement qui n'existent pas en réalité.
3. $L(R, m_0) \subset L(GC(R, m_0))$
   (Graphe de couvreture est une sur-approximation du comportement réel)

$N\#11$: Réseau parenthèses (expressions bien parenthésées)





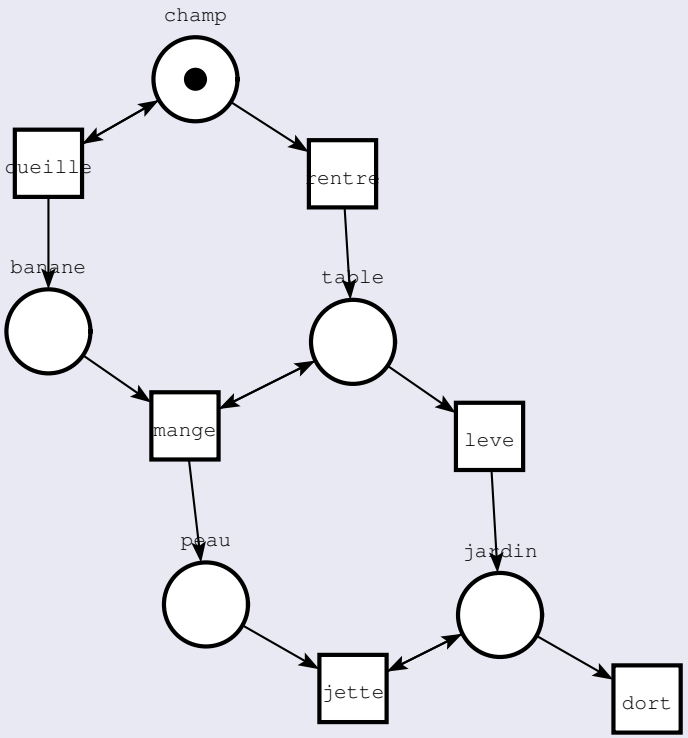


[]]] n'est pas bien parenthésée ($\notin L(R, m_0)$) mais est "reconnue" par le graphe de couverture ($\in L(GC(R, m_0))$)

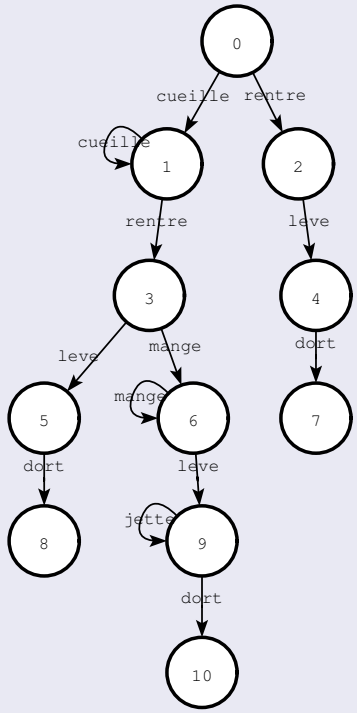
Les langages de Dick ne sont pas rationnels : impossible de les reconnaître par des automates à états finis

Journée d'un planteur de bananes

Réseau



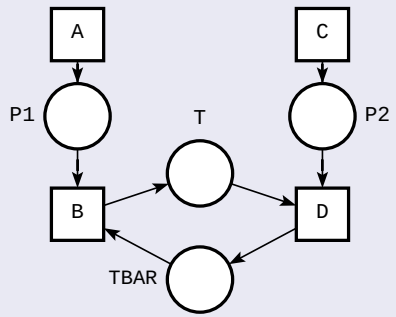
Graphe de Couverture



Outline

Introduction

Objectif



Etre capable d'obtenir certaines propriétés du réseau uniquement par une analyse de la structure/topologie du réseau.

Approche de la vérification

Elégante : pas d'exploration de l'espace d'états

Manuelle : calculs automatiques mais interprétation "manuelle"

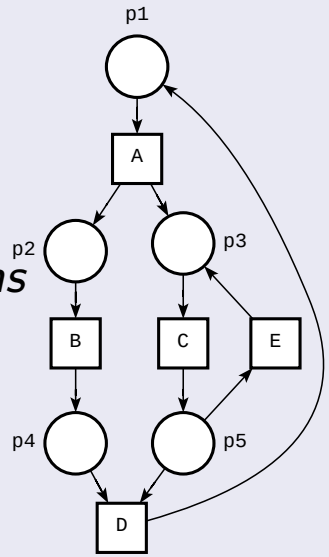
Puissante (possiblement) : analyse paramétrée

Décevante (possiblement) : peut échouer à monter une propriété vraie

Réseaux Place/Transition (rappels)

Définition RdP (rappel)

$R =< P, T, Pre, Post >$ où
- P, ensemble fini de *Places*
- T, ensemble fini de *Transitions*
- $Pre : P \times T \mapsto \mathbb{N}$
- $Post : P \times T \mapsto \mathbb{N}$



$$A \quad B$$
$$({}^\bullet A, A^\bullet) \quad ({}^\bullet B, B^\bullet)$$
$$\left(\begin{vmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{vmatrix}, \begin{vmatrix} 0 \\ 1 \\ 1 \\ 0 \\ 0 \end{vmatrix}\right) \quad \left(\begin{vmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{vmatrix}, \begin{vmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 0 \end{vmatrix}\right)$$

...

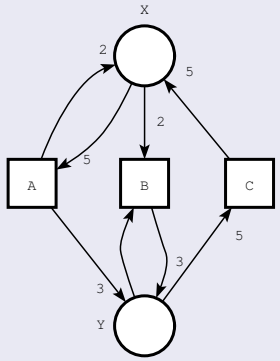
Matrices Pre et Post associées

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

Matrice d'incidence

Un autre exemple



Pre	*A*	*B*	*C*
	5	2	0
	0	1	5

Post	*A*	*B*	*C*
	2	0	5
	3	3	0

Matrice d'incidence

$C : P \times T \mapsto \mathbb{Z}$

C = Post - Pre

$C(p,t) = t^{\bullet}(p) -^{\bullet} {}^{\bullet}t(p)$

C	*A*	*B*	*C*
	-3	-2	5
	3	2	-5

Réseaux Purs

N est pur ssi $t^{\bullet}(p) \times^{\bullet} {}^{\bullet}t(p) = 0 \quad \forall t, \forall p$

Prop : Si N est pur, Il est possible de retrouver N à partir de C.
${}^{\bullet}t(p) = -Max(0, -C(p,t))$ et $t^{\bullet}(p) = Max(0, C(p,t))$

Image commutative d'une séquence de transitions

–

– : $T^* \mapsto \mathbb{N}^T$ définie par $\bar{\sigma}(t) = |\sigma|_t$
où $|\ \ |$: $T^* \times T \mapsto \mathbb{N}$ est définie par

$$|\epsilon|_t = 0 \qquad \text{et} \qquad |u.\sigma|_t = \begin{cases} 1 + |\sigma|_t & \text{si } u = t \\ |\sigma|_t & \text{sinon} \end{cases}$$

Exemple : $T = \{t1, t2, t3, t4\}$

$\sigma_1 = t1.t2.t2.t3$ $\quad |\sigma_1|_{t1} = |\sigma_1|_{t3} = 1$, $|\sigma_1|_{t2} = 2$, $|\sigma_1|_{t4} = 0$ $\quad \bar{\sigma_1} = \begin{vmatrix} 1 \\ 2 \\ 1 \\ 0 \end{vmatrix}$

$\sigma_2 = t4.t2.t1.t3$ $\quad |\sigma_1|_t = 1 \ \ \forall t \in T$ $\quad \bar{\sigma_1} = \begin{vmatrix} 1 \\ 1 \\ 1 \\ 1 \end{vmatrix}$

$\sigma_3 = t3.t2.t1.t2$ $\quad |\sigma_3|_{t1} = |\sigma_1|_{t3} = 1$, $|\sigma_1|_{t2} = 2$, $|\sigma_1|_{t4} = 0$ $\quad \bar{\sigma_3} = \begin{vmatrix} 1 \\ 2 \\ 1 \\ 0 \end{vmatrix}$

NB – n'est pas injective ($\sigma_3 \neq \sigma_1$ **ET** $\bar{\sigma_3} = \bar{\sigma_1}$)

Equation Fondamentale des réseaux de Petri

L'équation

Soient $m_1, m_2 \in A(R, m_0), \sigma \in T^*$: **Si** $m_1 \xrightarrow{\sigma} m_2$ **Alors** $m_2 = m_1 + C.\bar{\sigma}$

Réciproque **FAUSSE**

(1) on travaille avec $\bar{\sigma}$ et non σ

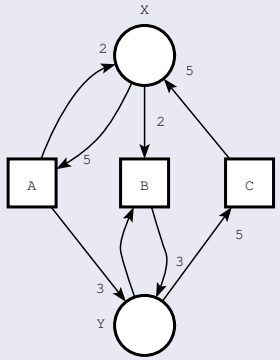
(2) on raisonne avec C et non avec *Pre* & *Post*

Corollaires

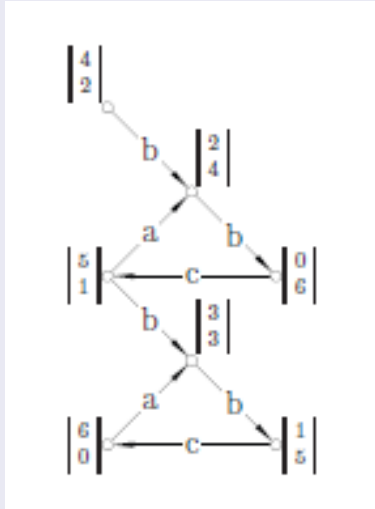
1. $\forall m \in G(R, m_0), \exists \sigma \in T^* \; m = m_0 + C.\bar{\sigma}$
   fondement de l'analyse structurelle
2. Soient $m_1, m_2 \in A(R, m_0)$ et $\sigma \in T^*$ tels que $m_2 = m_1 + C.\bar{\sigma}$
   Si $\exists p \in P : m_2(p) < 0$ Alors NON $(m_1 \xrightarrow{\sigma})$

Exemple d'application

Equation RdP : Si $m_1 \xrightarrow{\sigma} m_2$ **Alors** $m_2 = m_1 + C.\bar{\sigma}$



C	A	B	C
	-3	-2	5
	3	2	-5



Si $S = BBCA$ alors $\bar{S} = \begin{vmatrix} 1 \\ 2 \\ 1 \end{vmatrix}$ et $C.\bar{S} = \begin{vmatrix} -3 & -2 & 5 \\ 3 & 2 & -5 \end{vmatrix} \otimes \begin{vmatrix} 1 \\ 2 \\ 1 \end{vmatrix} = \begin{vmatrix} -2 \\ 2 \end{vmatrix}$

Ici $\begin{vmatrix} 4 \\ 2 \end{vmatrix} \xrightarrow{BBCA} M'$ avec $M' = \begin{vmatrix} 4 \\ 2 \end{vmatrix} + \begin{vmatrix} -2 \\ 2 \end{vmatrix} = \begin{vmatrix} 2 \\ 4 \end{vmatrix}$

De même, $\begin{vmatrix} 5 \\ 1 \end{vmatrix} \xrightarrow{BBCA} M''$ avec $M'' = \begin{vmatrix} 5 \\ 1 \end{vmatrix} + \begin{vmatrix} -2 \\ 2 \end{vmatrix} = \begin{vmatrix} 3 \\ 3 \end{vmatrix}$

Analyse structurelle/statique

Exploitation du corollaire de l'équation fondamentale des RdP

$\forall m \in G(R, m_0), \exists \sigma \in T^* : m = m_0 + C.\bar{\sigma}$

Principe : éliminer une des inconnues

- Chercher les applications F telle que $F.C = 0$
  $\longmapsto F.m = F.m_0$ *Invariants linéaires de Place*
- Chercher les σ vérifiant $C.\bar{\sigma} = 0$
  $\longmapsto m_0 \xrightarrow{\sigma} m$ avec $m_0 = m$ *Invariants linéaires de Transitions*

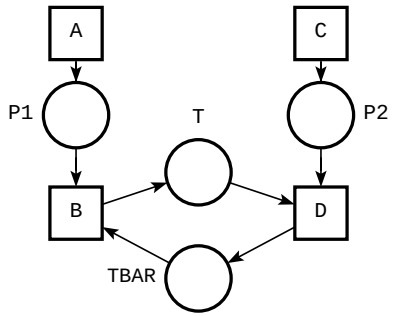
Retour sur la matrice d'incidence C

$C \in \mathbb{Z}^{P \times T}$

- Image de C, $Im(C) = \{M \in \mathbb{Z}^P : \exists \sigma \in \mathbb{Z}^T : C.\bar{\sigma} = M\}$
- Noyau de C, $Ker(C) = \{\sigma \in \mathbb{Z}^T : C.\bar{\sigma} = 0\}$

Prop : $A(N, m_0) \subseteq Im(C) \quad (\forall m_0 \in \mathbb{N}^P)$

Invariants de Place (Intuition)



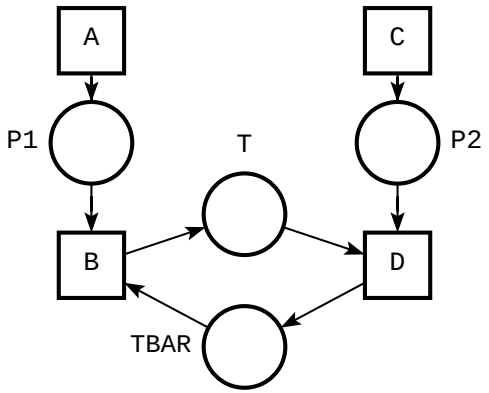
Focus sur les places T et $TBar$

1. A et C ne sont pas connectées aux places T et $TBar$
   Pour $x \in \{A, C\}$, Si $m \xrightarrow{x} m'$
   alors $m'(T) = m(T)$ et $m'(TBar) = m(TBar)$
2. Si $m \xrightarrow{B} m'$ alors $m'(T) = m(T) + 1$ et $m'(TBar) = m(TBar) - 1$
3. Si $m \xrightarrow{D} m'$ alors $m'(T) = m(T) - 1$ et $m'(TBar) = m(TBar) + 1$

donc pour $t \in \{B, D\}$

Si $m \xrightarrow{t} m'$ alors $m(T) + m(TBar) = m'(T) + m'(TBar)$

Invariants de Place (Intuition)



Bilan

1) A, C laissent $m(T)$ et $m(TBar)$ invariants
2) & 3) B, D laissent la **Somme** $m(T) + m(TBar)$ invariante
$\Rightarrow m(T) + m(TBar)$ est invariante

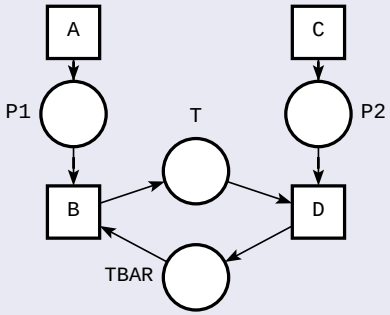
Conclusion

$\forall m \in A(R, m_0) : m(T) + m(TBar) = m_0(T) + m_0(TBar)$

Résutat très général puisqu'il est indépendant du marquage initial.
Impossible à obtenir en utilisant les approches exhaustives type arbre de couverture.

Invariants de Transition (Intuition)

Exemple



$$\begin{vmatrix} P_1 \\ P_2 \\ T \\ TBar \end{vmatrix} \overset{A.B.C.D}{\rightarrow} \begin{vmatrix} P_1 \\ P_2 \\ T \\ TBar \end{vmatrix}$$

La séquence A.B.C.D laisse le marquage invariant

Dans le détail ....

$$\begin{vmatrix} P_1 \\ P_2 \\ T \\ TBar \end{vmatrix} \overset{A}{\rightarrow} \begin{vmatrix} P_1+1 \\ P_2 \\ T \\ TBar \end{vmatrix} \overset{B}{\rightarrow} \begin{vmatrix} (P_1+1)-1 \\ P_2 \\ T+1 \\ TBar-1 \end{vmatrix} \overset{C}{\rightarrow} \begin{vmatrix} P_1 \\ P_2+1 \\ T+1 \\ TBar-1 \end{vmatrix} \overset{D}{\rightarrow} \begin{vmatrix} P_1 \\ (P_2+1)-1 \\ (T+1)-1 \\ (TBar-1)+1 \end{vmatrix}$$

Vérification

C	A	B	C	D
P_1	1	-1	0	0
P_2	0	0	1	-1
T	0	1	0	-1
$TBar$	0	-1	0	-1

$$\times \begin{vmatrix} 1 \\ 1 \\ 1 \\ 1 \end{vmatrix} = \begin{vmatrix} 0 \\ 0 \\ 0 \\ 0 \end{vmatrix}$$

Invariants de Place (P semi-flots)

Objectif Chercher les applications F tq $F.C = 0$ et par suite $F.m = F.m_0$

Propriéte : Soit F telle que $F.C = 0$

En posant $P = \{p_1, p_2 \dots p_n\}$ et en développant $F.m = F.m_0$ on obtient :

$$\sum_{j=1}^{j=n} f_j \times m(p_j) = \sum_{j=1}^{j=n} f_j \times m_0(p_j)$$

$$\sum_{j=1}^{j=n} f_j \times m(p_j) = Constante(m_0) \qquad (\forall m \in \mathbb{N}^P)$$

Propriéte : Soit F à coéfficients dans $\mathbb{N}$ telle que $F.C = 0$

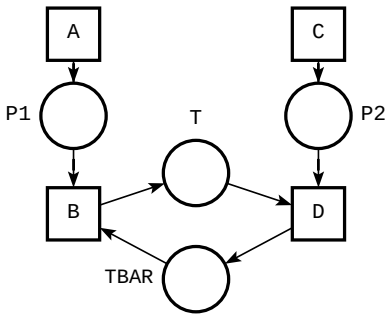
Si $f_i \neq 0$ alors la place p_i est bornée

$$f_i \times m(p_i) = Cste(m_0) - (\sum_{j \in [1,p] \setminus \{i\}} f_j \times m(p_j))$$

$$m(p_i) = 1/f_i \times (Cste(m_0) - \sum_{j \in [1,p] \setminus \{i\}} f_j \times m(p_j)$$

$$m(p_i) \leq m(m_0)/f_i \textbf{ Et } p_i \textbf{ est bornée}$$

Invariants de Place du Producteur/Consommateur



$$\begin{vmatrix} 1 & -1 & 0 & 0 \\ 0 & 0 & 1 & -1 \\ 0 & 1 & 0 & -1 \\ 0 & -1 & 0 & -1 \end{vmatrix}$$

On résoud $\begin{vmatrix} F_{P_1} & F_{P_2} & F_T & F_{TBar} \end{vmatrix} \times C = \begin{vmatrix} 0 \\ 0 \\ 0 \\ 0 \end{vmatrix}$

$F.C = 0$ ssi $\begin{vmatrix} F_{P_1} & F_{P_2} & F_T & F_{TBar} \end{vmatrix} = \lambda. \begin{vmatrix} 0 & 0 & 1 & 1 \end{vmatrix}$

Comme F_T & $F_{TBar} \neq 0$ on a T et $TBar$ bornées

De plus, en prenant $m_0 = \begin{vmatrix} 0 \\ 0 \\ 0 \\ k \end{vmatrix}$ et en développant $F.m = F.m_0$ on obtient :

$m(T) \leq K$ & $m(T') \leq K \quad (\forall m \in G(R, m_0))$

Composante Conservative, Répétitive

Composante Conservative - Invariant de Place, P semi-flots

Def : $B \subseteq P$ est un invariant de Place ssi $\exists F \geq 0$ tq $||F|| = B$ et $F.C = 0$
où $||F|| = \{p_i \in P : F_i \neq 0\}$

Prop : Si B est un invariant de place, alors toute place p de B est bornée
réciproque fausse

Def : R est conservatif (consistent) si P est couvert par des invariants de place

Prop : Si R est conservatif/consistent alors R est borné

Composante Conservative - Invariant de Transitions, T semi-flots

Def : $S \subseteq T$ est un **invariant de Transition** ssi
$\exists s \in T^*$ telle que $||\sigma|| = S$ et $C.\bar{\sigma} = 0$

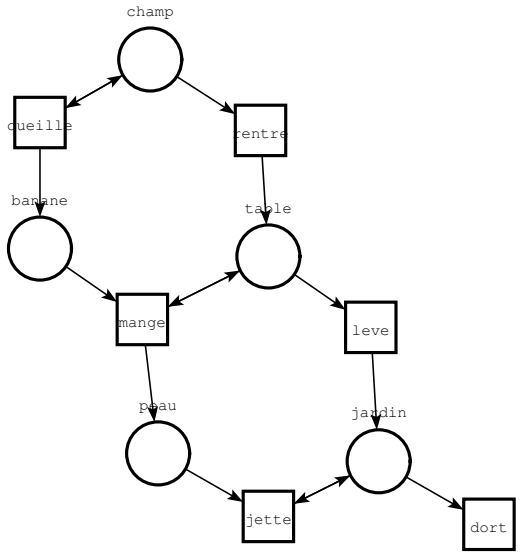
Def : R est répétitif (invariant) si T est couvert par des invariants de transition

Prop : Si R est vivant et borné est Alors R est répétitif

nb :

Infinité possible d'invariants. L'analyse permet d'en obtenir une base.

Exemple de réseau sans invariant de place



T semi-flot

Cueille.Mange.Jette est une séquence invariante

Pour que l'invariant de transitions soit **effectif** (soit franchissable) il faut au moins 3 planteurs ; un au champ, un à table et un dans le jardin.

Aucun P semi-flot

Pour m_0 donné et $m \in A(R, m_0)$ on a
$m(champ) + m(table) + m(jardin) \leq m_0(champ) + m_0(table) + m_0(jardin)$

Les places champ, table et jardin ne forment pas un invariant de place, pour autant les places champ, table et jardin sont bornées.

On sait par ailleurs que banane et peau sont non bornées et ne peuvent appartenir à un invariant de place.

Modélisation et vérification struturelle du Pb des lecteurs/écrivains

Enoncé

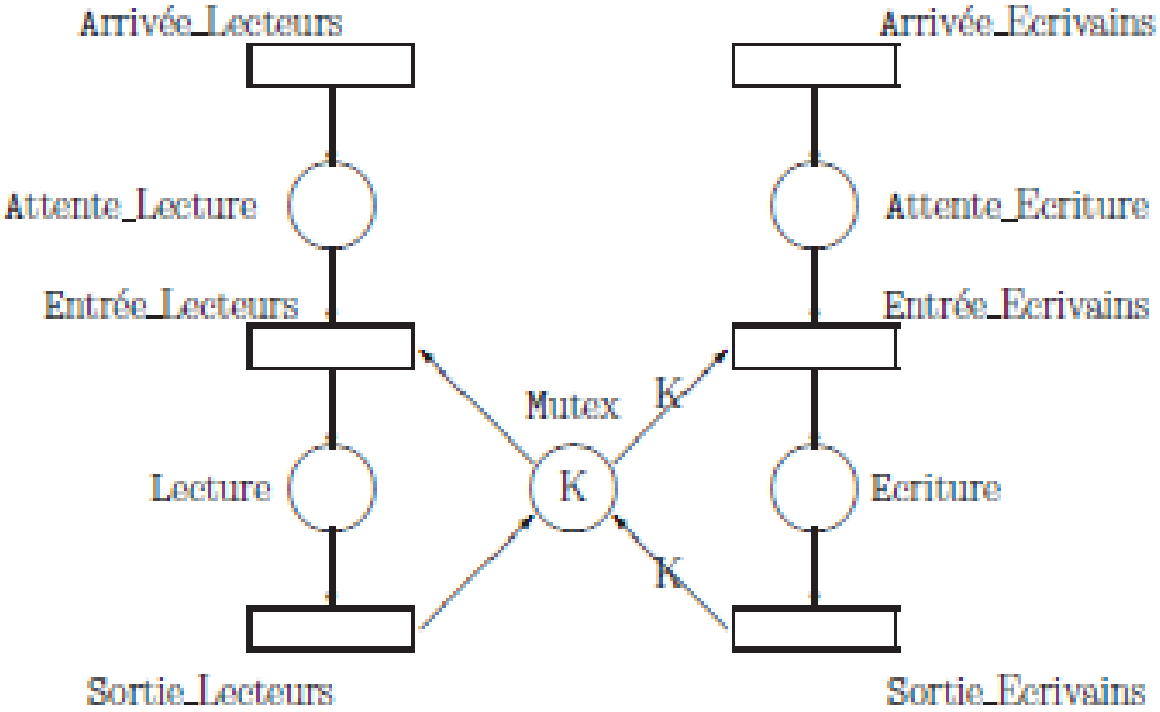
Lecteurs et écrivains en concurrence pour accéder à une (abstraction) de base de données. Modéliser un arbitre (contrôleur) permettant de synchroniser les lecteurs et les écrivains pour assurer les propriétés de sûreté ci-dessous

- C_1 : Pas plus de k lectures simultanées
- C_2 : Pas plus d'un écrivain dans la base
- C_3 : Pas de lecture et d'écriture simultanément dans la base

Aspects non traités :

- Vivacité/"Absence de Famine" : Toute requète (de lecture ou d'écriture) est satisfaite au bout d'un temps fini.
- Maintien de la cohérence des données (si une valeur est lue alors elle correspond à la dernière valeur écrite).

Un modèle possible



Modélisation paramétrée

Modèle de la base de données "générique" :

entier **positif** k est un paramètre du marquage et du **réseau**.

On suppose que la population de lecteurs et d'écrivains est infinie.

Légende

Les "composants"

A droite les lecteurs, à gauche les écrivains et au centre le contrôle.

Les "lieux"

- une salle de lecture et d'une salle d'écriture,
- deux salles d'attente : l'une pour les lecteurs et l'autre pour les écrivains.

Le "contrôle" : Mutex synchronise lecteurs et écrivains.

Expression dans le modèle des propriétés attendues : $\forall m \in A(R, m_0)$

- $C_1 : m(Lecture) \leq k$ — *Pas plus de k lectures simultanées*
- $C_2 : m(Ecriture) \leq 1$ — *Pas plus d'un écrivain dans la base*
- $C_3 : m(Ecriture) \times m(Lecture) = 0$ — *Pas de lecture et d'écriture simultanées*

Remarque Modélisation/Vérification

On fait le modèle pour l'analyser/vérifier. On doit avoir identifier (avant) les propriétés que l'on voudra vérifier. On doit donc s'assurer, au plus tôt, que l'on est déja capable de les exprimer dans/avec le modèle.

Si on est pas capable de les exprimer (simplement) alors on ne pourra pas les vérifier (sûrement).

Calcul et exploitation des P semi-flots

On résoud $F.C = 0$
En ayant pris $P = [Lecture, Mutex, Ecriture, Attente\_Lecture, Attente\_Ecriture]$
On obtient $F = \lambda(1, 1, K, 0, 0)$

Sachant que $K > 0$ on en déduit que les places Lecture, Ecriture et Mutex sont bornées.

De plus, pour $m_0 = \left| \begin{array}{c} 0 \\ K \\ 0 \\ 0 \\ 0 \end{array} \right|$ et $m \in A(R, m_0)$ on a

$$F.m = (1, 1, K, 0, 0) \times \left| \begin{array}{l} m(Lecture) \\ M(Mutex) \\ M(Ecriture) \\ M(Attente\_Lecture) \\ m(Attente\_Ecriture \end{array} \right| = F.m_0 = 0$$

et finalement $m(Lecture) + m(Mutex) + k \times m(Ecriture) = K$

Vérification structurelle des propriétés

1. En isolant $m(Lecture)$ dans l'invariant, on obtient :

$$m(Lecture) = K - (m(Mutex) + K \times m(Ecriture))$$

Comme un marquage est une application à valeurs dans les entiers naturels, on en déduit $C_1 : m(Lecture) \leq K$.

2. $C_2 : m(Ecriture) \leq 1$ peut être obtenue de facon identique en isolant $m(Ecriture)$ dans l'invariant et en se souvenant que K est strictement positif.

3. En prenant la négation de C_3, on obtient

$$m(Lecture) > 0 \text{ et } m(Ecriture) > 0$$

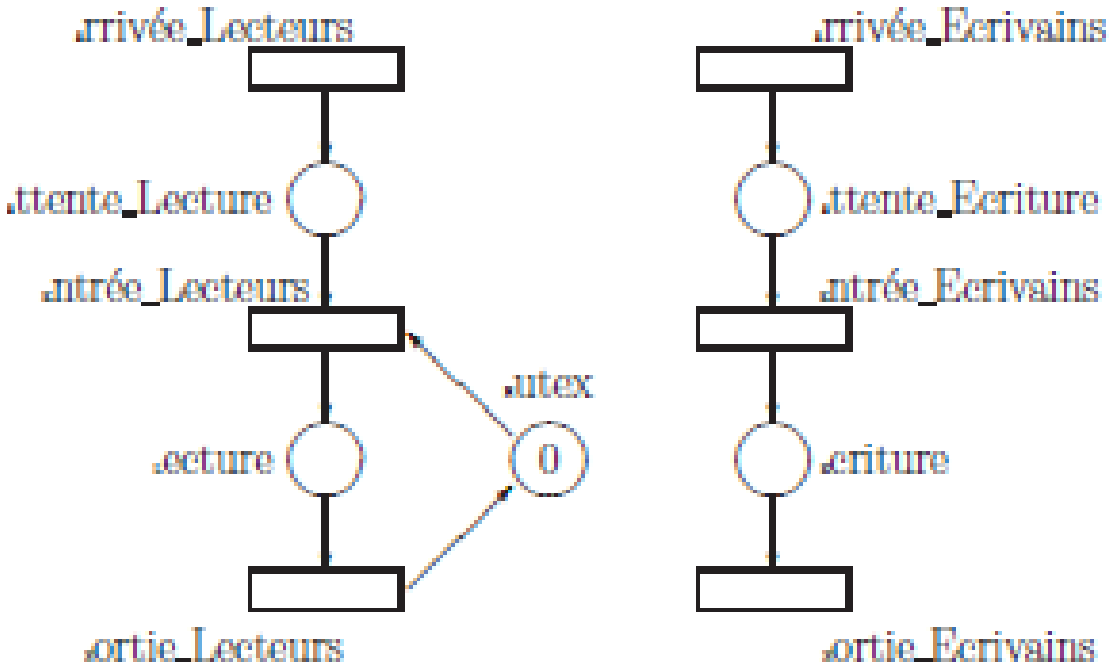
On a donc $m(Lecture) + K \times m(Ecriture) > K + 1$

D'un autre côté, l'invariant nous assure que :

$$m(Lecture) + K \times m(Ecriture) \leq k$$

Comme $K > 0$, on aboutit ainsi à la contradiction et C_3 est validée.

Quid de $K = 0$



Bilan

- L'analyse structurelle ne permet plus d'assurer C_2 et C_3.
- La place Mutex n'est plus une pré-condition de la transition Entrée_Ecrivains ; L'arc correspondant disparaît car il est valué par 0 et le nombre d'écritures simultanées devient non borné.
- La lecture devient impossible puisque le marquage initial de la place Mutex est nul.

Bilan Analyse structurelle

Analyse paramétrée

Possibilité de pouvoir travailler avec l'analyse structurelle sur un réseau paramétré.

Les propriétés attendues ont été établies non pas pour un réseau marqué mais pour toute une famille (infinie) indexée par $K > 0$ de réseaux marqués.

Une approche exhaustive (construction du graphe d'accessibilité, arbre de couverture de Karp et Miller) n'aurait pu être conduite puisque nous aurions du auparavant fixer une valeur pour le paramètre K.

Vivacité/Sûreté

L'analyse structurelle est une approche séduisante pour la vérification des propriétés dites de "Sûreté"

"Rien de mauvais ne peut arriver"

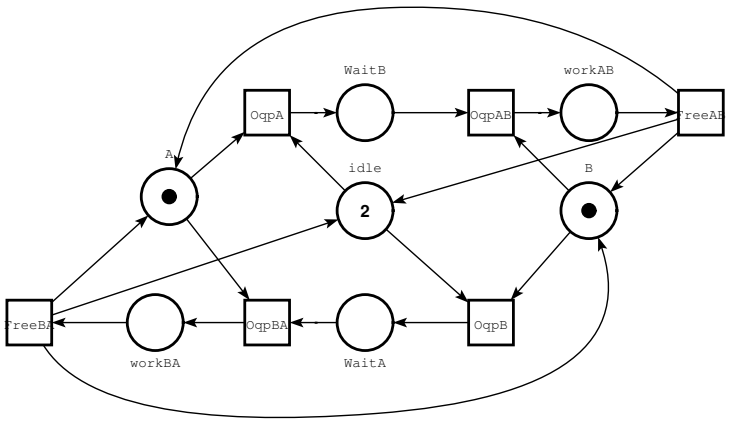
L'analyse structurelle n'appporte (en général) pas de réponse pour la vérification des propriétés dites de "Vivacité"

"Quelque chose de bon doit arriver"

Analyse structurelle avec TINA

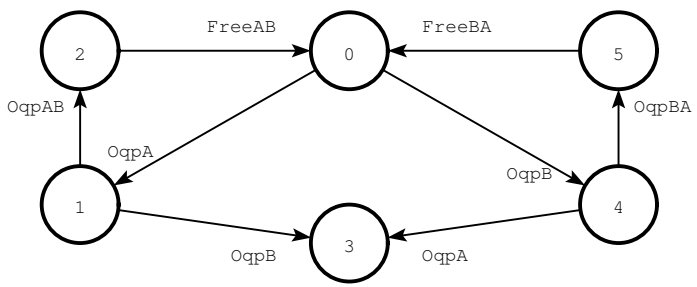
Enoncé

Deux processus (banalisés) partagent deux ressources (A et B). L'un des processus se procure d'abord A (OqpA), puis se procure B (OqpAB), il travaille (WorkAB) puis il retourne au repos en libérant (FreeAB) les deux ressources. L'autre processus procède dans l'ordre inverse : obtention de B puis de A.



Exploration de l'espace d'états

```
0 : A B idle*2
1 : B WaitB idle
2 : idle workAB
3 : WaitA WaitB
4 : A WaitA idle
5 : idle workBA
```



Analyse structurelle avec TINA : P Semi-flots

(rappel)

```
0 : A B idle*2
1 : B WaitB idle
2 : idle workAB
3 : WaitA WaitB
4 : A WaitA idle
5 : idle workBA
```

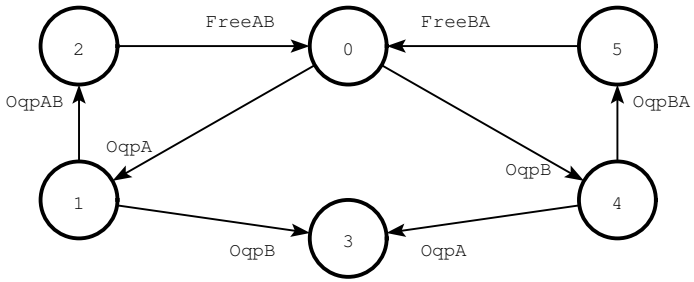
```
P-SEMI-FLOWS GENERATING SET

invariant
A WaitB workAB workBA
WaitA WaitB idle workAB workBA
B WaitA workAB workBA
```

Interprétation/Remarques

- Invariant (conservatif) : Réseau structurellement borné
- Ensemble (infini) des invariants est généré par une base de 3 invariants
  $m(A) + m(WaitB) + m(workAB) + m(workBA) = cste_a(m_0)$
  $m(WaitA)+m(WaitB)+m(idle)+m(workAB)+m(workBA) = cste_b(m_0)$
  $m(B) + m(WaitA) + m(workAB) + m(workBA) = cste_c(m_0)$
- L'exclusion mutuelle peut être prouvée en utilisant Inv_1 (ou Inv_3)

Analyse structurelle avec TINA : T Semi-flots




```
T-SEMI-FLOWS GENERATING SET

consistent

FreeAB OqpA OqpAB

FreeBA OqpB OqpBA
```

Interprétation/Remarques

- Consistent (i.e., répétitif)
  chaque transition appartient à un invariant de transitions
- Ensemble (infini) des invariants est généré par une base de 2 invariants
  FreeAB OqpA OqpAB
  FreeBA OqpB OqpBA

  Interpétation :
  Si $M - OqpA.OqpAB.FreeBA- >$
  alors $M - OqpA.OqpAB.FreeBA- > M$
  (1) l'orde des transitions dans la séquence n'est pas fourni !
  (2) un tel marquage M n'est pas forcément accessible.
- L'existence de ces invariants laisse présager la possibilité de famine !
  C'est bien le cas ici !

En conclusion

Intérêt des RdP

- Formalisme Rigoureux (sémantique précise)
- Expressivité
  + Compact (vs machines à états)
  + Concepts : //, Coopération, Compétition, Synchronisation, ...
  + Dualité Etat/Evénement
- Possibilités d'Analyse : Structurelle, Exhaustive
- Représentation Graphique

Aspects non traités

- Réductions
- Extensions des RdP
  - arcs inhibiteurs, priorités (voir tp)
  - Données (Réseaux colorés, Réseaux Prédicat/transitions)
  - Temps (temps réel)
  - Stochastique (performances)
- + généralement (hors rdp) : techniques de vérification