Modélisation de systèmes dynamiques
Spécifications formelles

Amal El Fallah Seghrouchni

Amal.Elfallah@lip6.fr

Carl Adam Petri

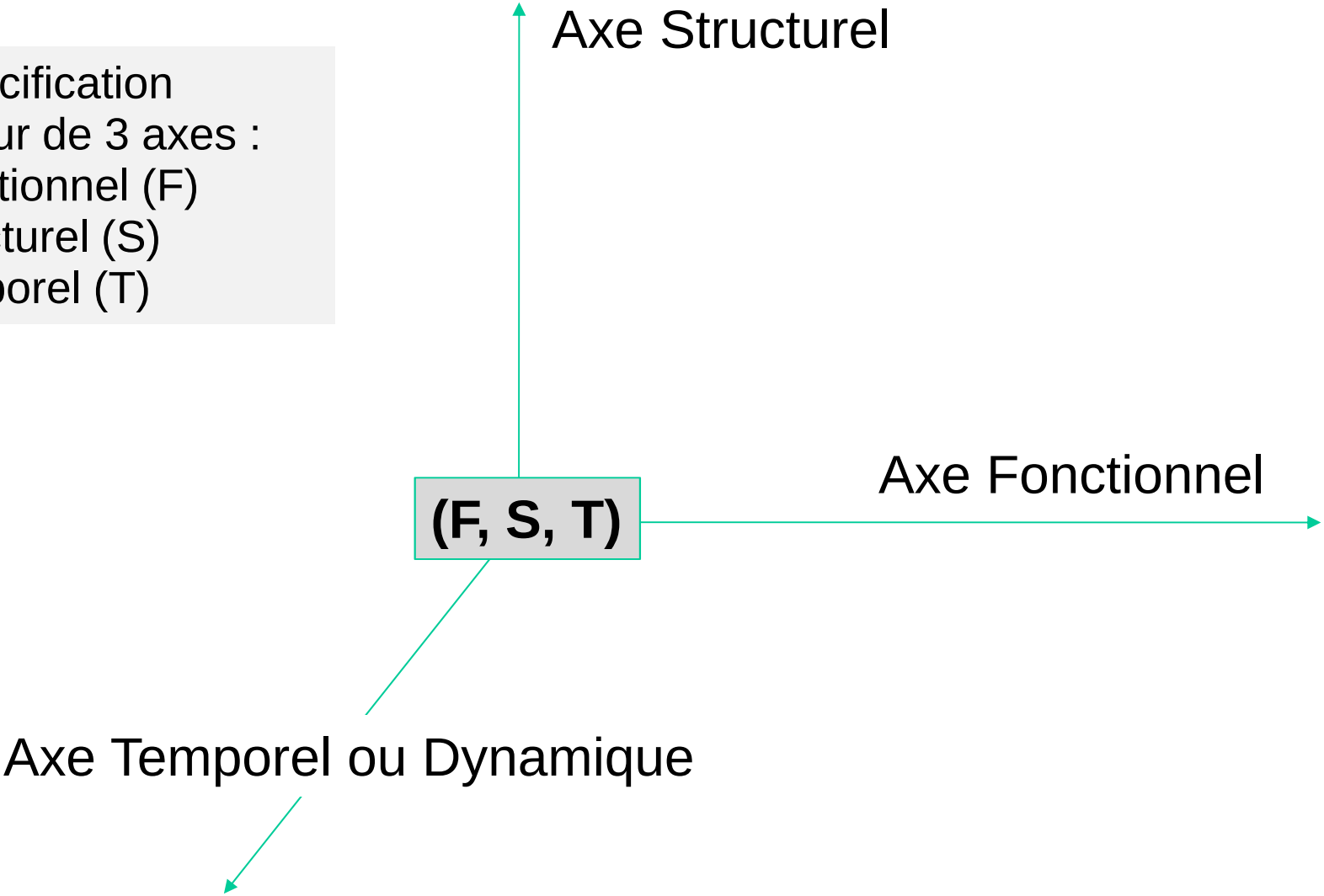
Plan

- Introduction
- Spécification d’un système dynamique
- Spécification formelle
- Réseaux de Petri (C.A. Petri, 1962)

Espace de Spécification d'un SD

Espace de spécification
-structuré autour de 3 axes :

- Axe Fonctionnel (F)
- Axe Structurel (S)
- Axe Temporel (T)



Aspects fonctionnels et structurels

- Aspect fonctionnel : « à quoi ça sert ? »
  - Exemple : le conducteur qui utilise une voiture la voit comme un objet permettant de se déplacer
- Aspect structurel : « quelle structure ? »
  - Exemple : le garagiste voit la voiture comme un ensemble de pièces qui interagissent et produisent un mouvement.

Remarques

- Il est important de savoir si l'on observe ou conçoit un système et ses sous-systèmes dans leurs aspects fonctionnels ou structurels.
- En effet, un système peut être étudié (ou conçu) dans son aspect structurel alors que certains des sous-systèmes qu'il contient, communs à d'autres système, ne seront approchés que d'une manière fonctionnelle (et inversement)
  - *Le garagiste peut changer les roues d'une voiture sans connaître le fonctionnement du moteur.*

Vues complémentaires du système Fonctionnelle, Structurelle et Temporelle

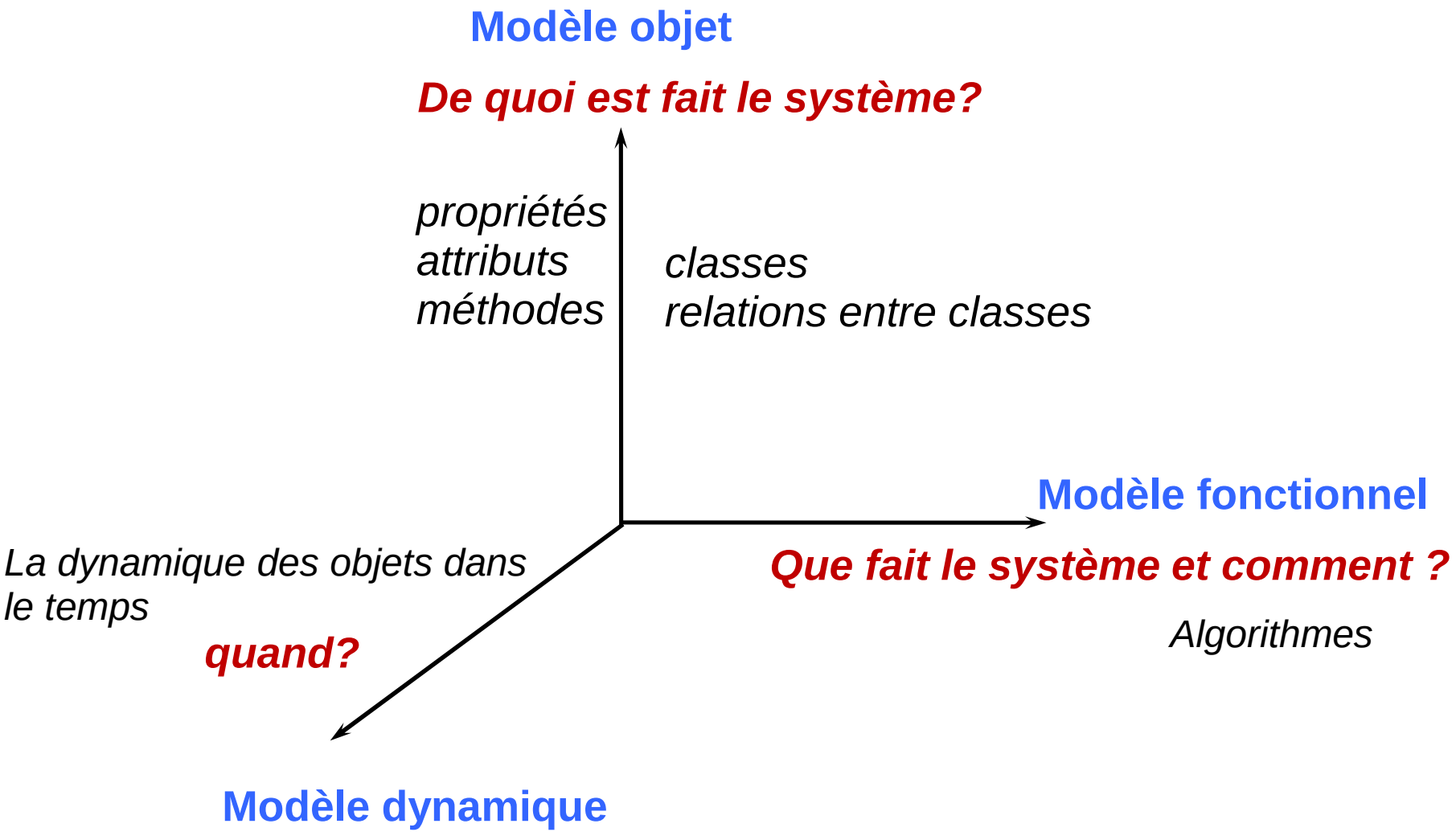
La complexité des systèmes oblige à étudier chacune de ces vues indépendamment des 2 autres.

Exemples de méthode sur chacun des axes :

- l'axe Fonctionnel : SA (Structured Analysis)
  - Les diagrammes associés illustrent les fonctions du système
- l'axe Structurel : E/R (Entités/Relations)
  - Le modèle E/R illustre les données du système
- l'axe Temporel :
  - Réseaux de Petri, automates temporisés, Statecharts, Diagrammes Etats/transitions, etc.

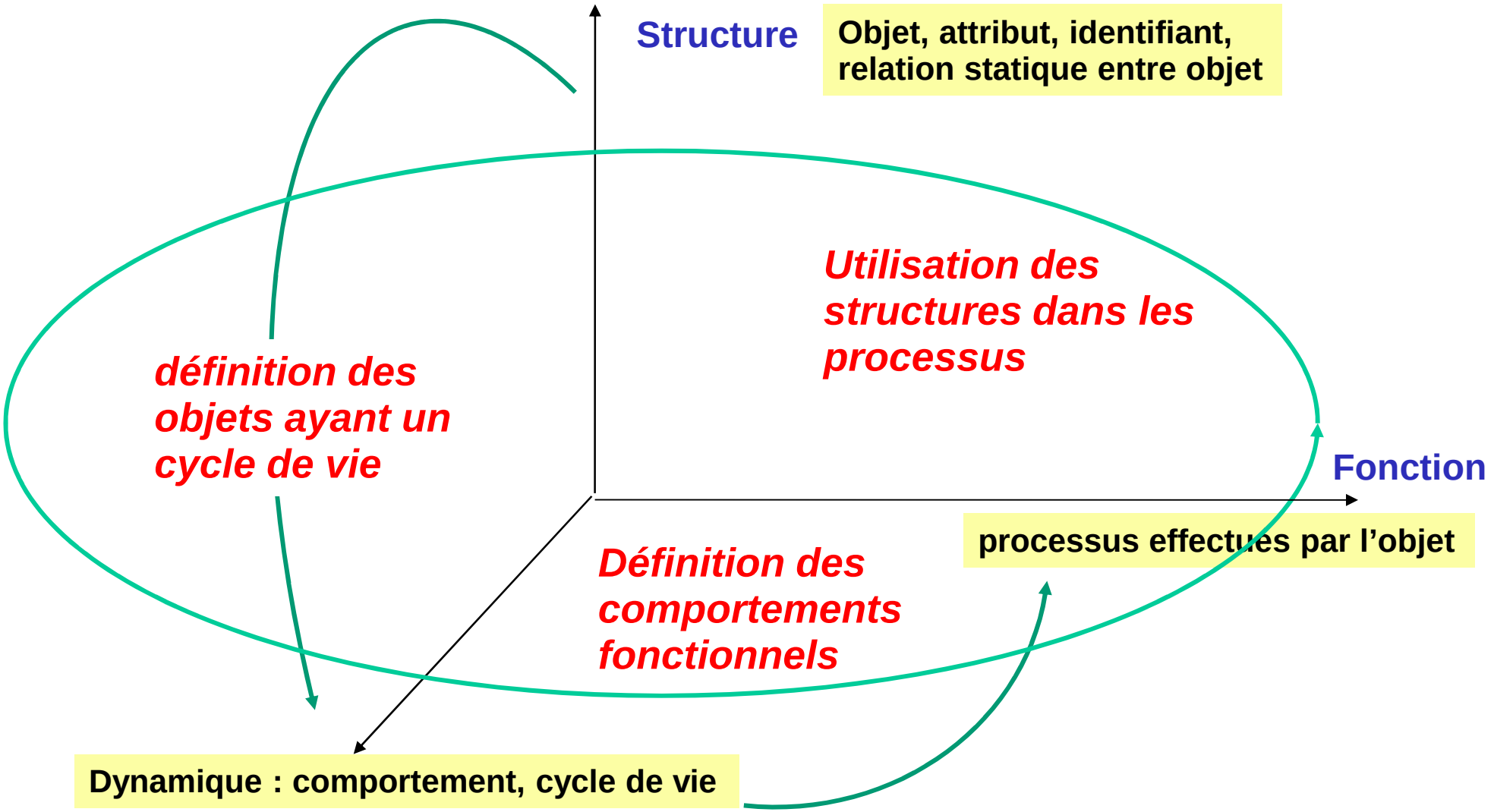
Ces formalismes permettent de décrire des modèles pour spécifier et analyser la dynamique (comportement) du système

Exemple : Axes de modélisation Objet



Application aux méthodes OO

Analyse Orientée Objet

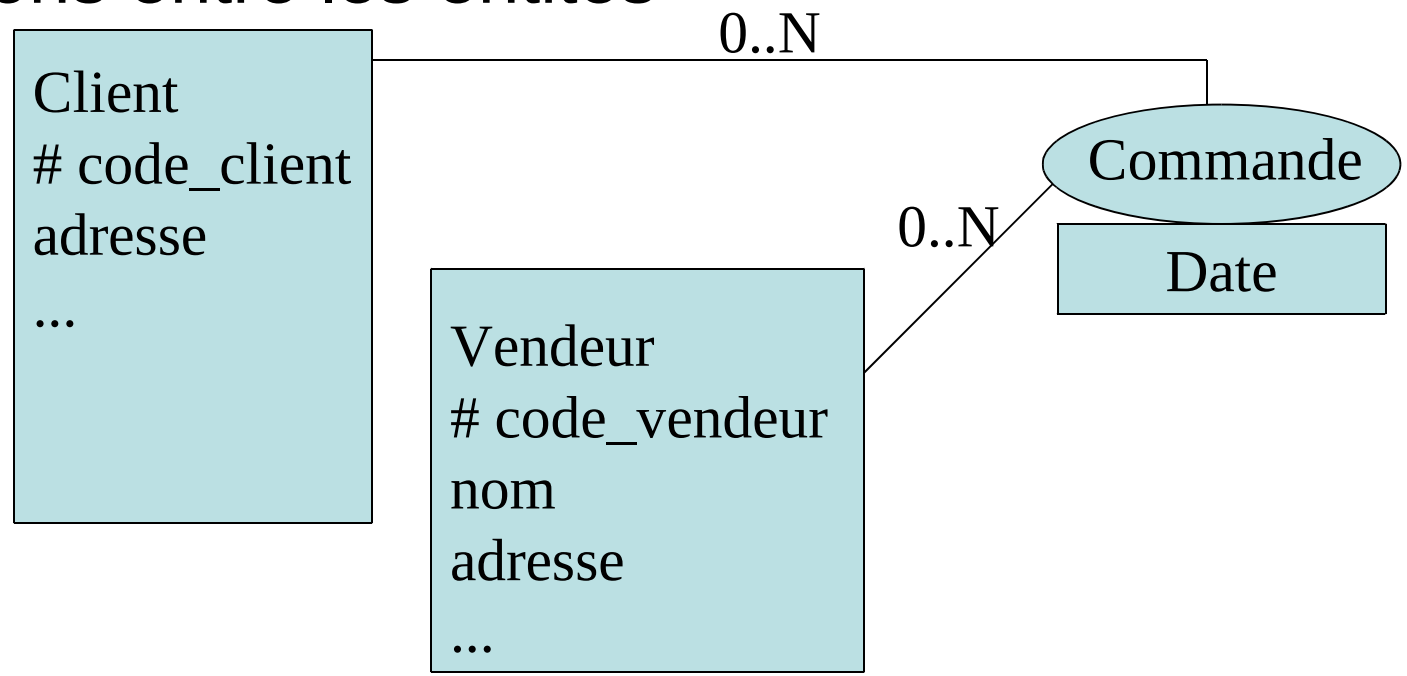


Analyse OO - Définition des objets

- A chaque cycle de vie d'une entité du monde réel correspondent des états de l'objet du modèle d'analyse
- A chaque événement du monde réel correspond un événement agissant sur les objets du modèle d'analyse
- A chaque modification des entités du monde réel correspond un processus activé dans un état des objets du modèle d'analyse

Exemple Entité/Relation

- Entité : Représente un ensemble, une collection, des objets du monde réel qui joue un rôle dans le système à étudier.
- Une entité peut être décrit par plusieurs attributs
- Relation : représente un ensemble de liens ou associations entre les entités



Problème de cohérence ...

Questions

Comment arriver à des spécifications cohérentes entre elles sur les 3 axes ?

Besoin de méthodes

- *Une méthode est un processus opératoire formel*
  - *Elle définit un ordre logique dans lequel les tâches doivent être réalisées pour atteindre un but défini*
  - *Elle spécifie l'inventaire, la nature et le contenu des tâches ainsi que l'ordre d'exécution et les résultats escomptés de la tâche.*

Méthodes semi-formelles

- Une méthode semi-formelle
  - Un langage (parfois graphique) + syntaxe précise + sémantique non précise (ou inexistante)
  - Divers outils d'analyse
  - Exemple : SADT, SA-RT, Merise, OMT, **UML**
- Elles sont utiles pour défraîchir le problème
  - En particulier en phase d'analyse
- Impossibilité de raisonner formellement sur le système à concevoir
  - Ne permettent pas la preuve.

Méthodes formelles

- Une méthode formelle
  - Un langage formel :
    - syntaxe précise + sémantique précise (spécifications formelles)
  - Système de preuve ou de raisonnement formel

- Il existe différents types de méthodes formelles
  - la classification repose sur les spécifications formelles sous-jacentes

Classification des méthodes formelles

- Spécifications orientées *propriétés*
  - description des *données* dans un langage permettant d'énoncer les propriétés attendues du système :
    - logique
    - logique temporelle
    - types abstraits algébriques

Classification des méthodes formelles

- Spécifications orientées *modèles*
  - construction d'un système à partir d'*objets* fondamentaux pré-établis
    - Réseaux de Petri (systèmes concurrents, communicants et distribués)
    - Algèbre de processus : CCS, CSP, Pi-calcul, calcul des « ambients », etc.

Classification des méthodes formelles

- Méthodes *hybrides*
  - VDM, Z, B (systèmes séquentiels) :
    - théorie des ensembles et système pré-, post-conditions
    - très utilisées dans l'industrie (malgré leur manque de définitions formelles)
  - LOTOS (Language of Temporal Ordering)
    - Algèbre de processus et types abstraits algébriques
  - Réseaux de Petri Algébriques
    - Réseaux de Petri et types abstraits algébriques

Intérêt : développement formel

Principe :

- transformation **systématique** des spécifications en programme
- Ces transformations utilisent des lois prédéfinies

  - Vérification : s'assurer que le système est correct par rapport à des propriétés

  - Validation : s'assurer que le système est correct par rapport aux spécifications

  - Raisonnement formel : appliquer un système formel à une spécification

Raisonnement formel

Exemples :

- Raffinement de spécification
- Vérification des propriétés d’un système
- Validation par vérification
- Preuve de théorèmes
- Analyse d’un système (représenté par une machine à états) par rapport à des propriétés (Model checking)

Spécification formelle orientée modèles

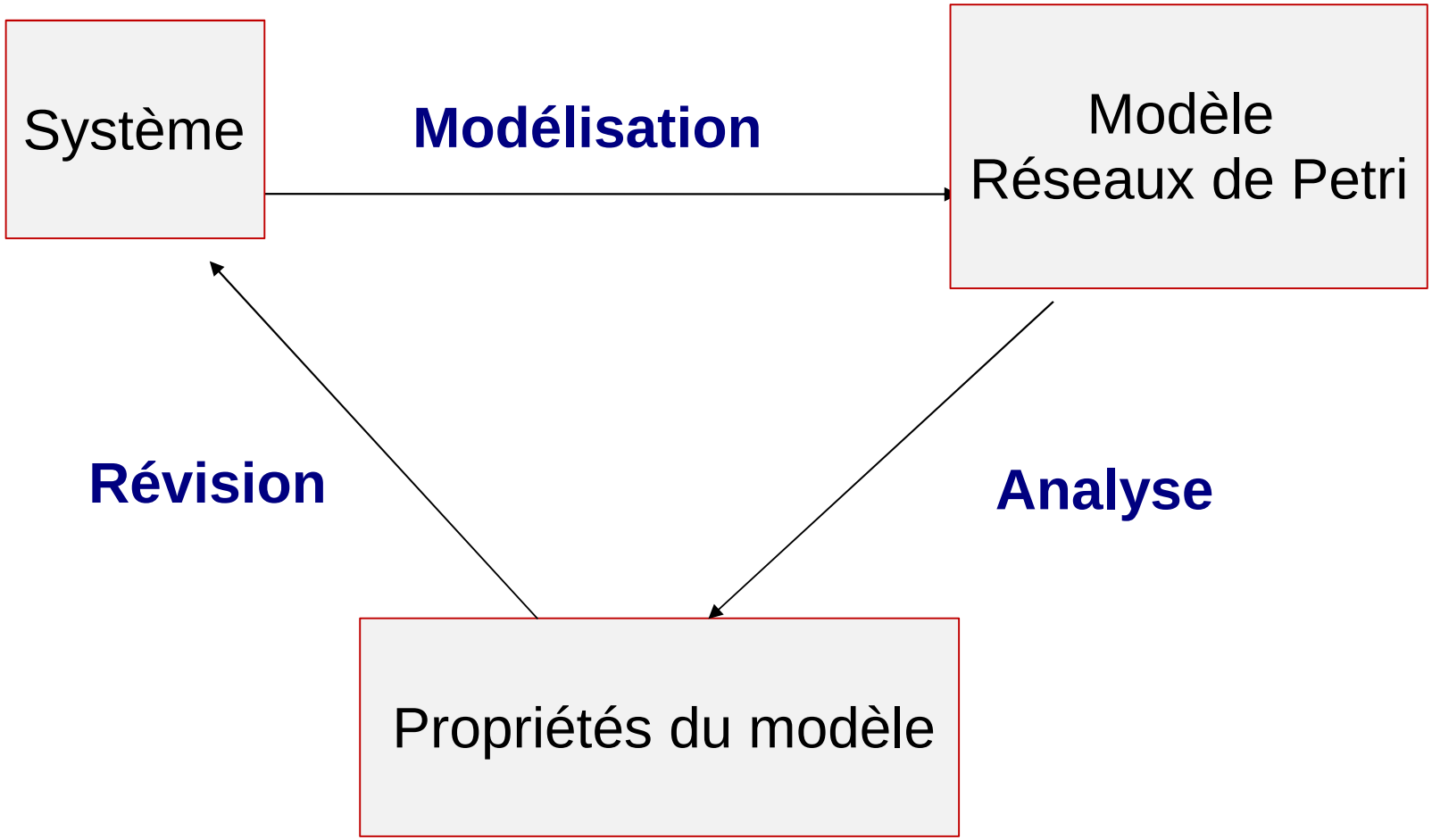
Les réseaux de Petri

Spécifications formelles au moyen de RdP

- Principe :
  - Données -> états
  - Construction d'un modèle du système
  - Définition des propriétés du système
  - Raisonnement sur le fonctionnement du système
  - Outils utilisés : mathématiques, logiques
- Intérêt :
  - Modélisation
  - Propriétés d'un modèle RdP
  - Vérification des propriétés

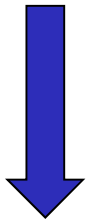
Modélisation RdP

Quels systèmes ? Discrets, Dynamiques et concurrents



Exemple

Boucle : le robot R ne cesse de prendre et de déposer des cubes



Définir les événements et les conditions du système.

Concepts de base : événements

- Evénements
  - Actions se déroulant dans le système
  - Déclenchement d'un événement dépend de l'état du système
  - Un état du système peut être décrit comme un ensemble de conditions
- Exemples
  - un cube arrive (e1)
  - le robot saisit le cube (e2)
  - le robot dépose le cube (e3)

Concepts de base : Conditions

- Conditions ou états du système
  - Une condition est un prédicat ou une description d'un état du système
  - Une condition est vraie ou fausse
- Exemples
  - Le robot est au repos (c1)
  - Un cube est en attente (c2)
  - Un cube est en cours de déplacement (c3)
  - Le cube a été déposé (c4)

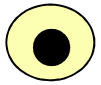
Concepts de base

- Déclenchement, pré-condition, post-condition
  - Les conditions nécessaires au déclenchement d'un événement sont les pré-conditions de l'événement
  - Le déclenchement d'un événement valide les post-conditions et peut invalider les pré-conditions de l'événement

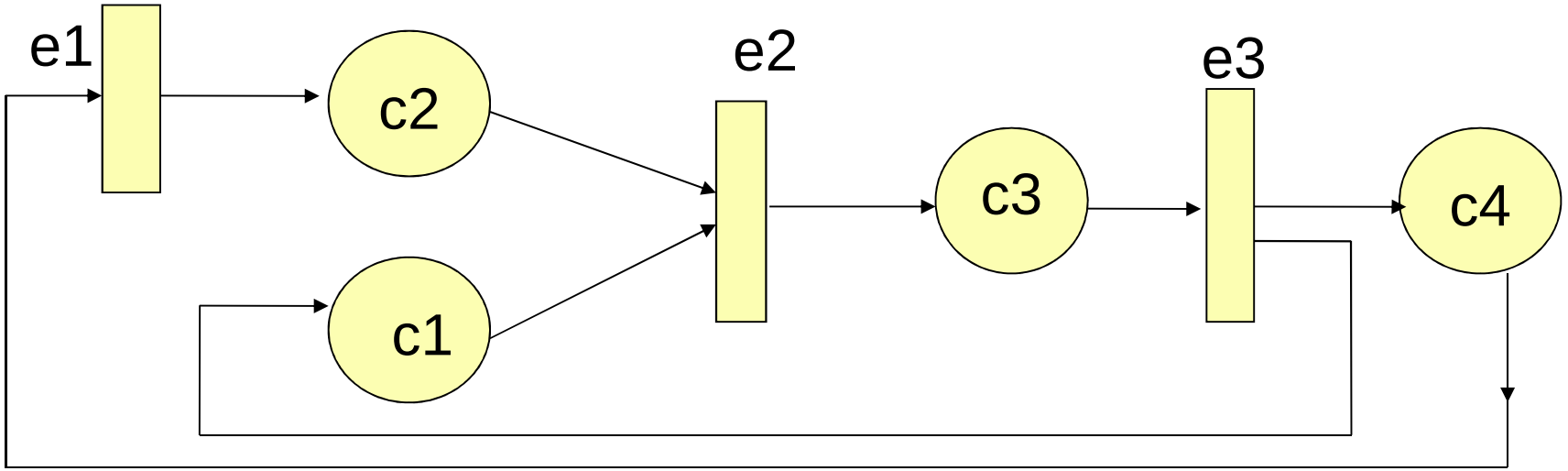
Modélisation du système

- Condition : place
  - robot est au repos (c1)
  - cube est en attente (c2)
  - cube est en cours de déplacement (c3)
  - cube a été déposé (c4)

Jeton = condition vraie

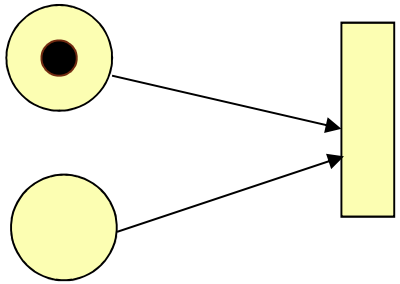


- Evénement : transition
  - un cube arrive (e1)
  - le robot saisit le cube (e2)
  - le robot dépose le cube (e3)

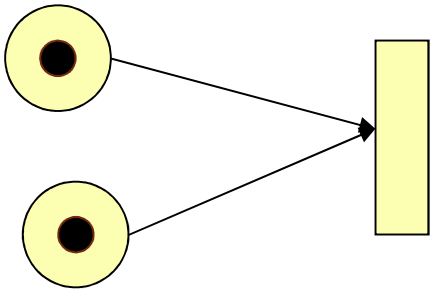


Boucle : le robot ne cesse de prendre et de déposer le cube

Condition de tir d’une transition

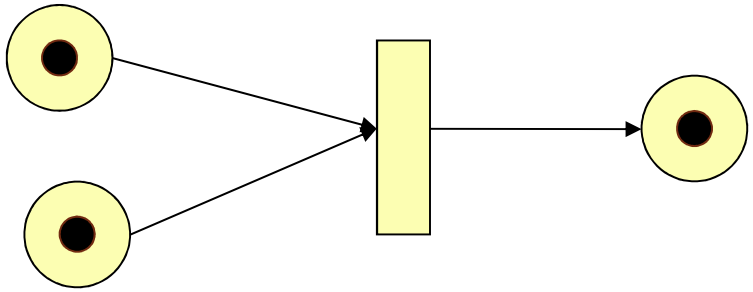


Non franchissable

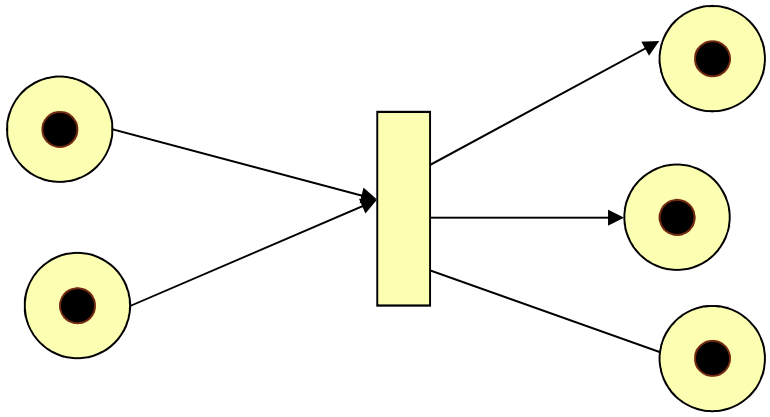


Franchissable

Déclenchement, franchissement ou tir d'une transition



Franchissement



Franchissement

Lors du franchissement, les jetons correspondants sont consommés des places en entrée.

Après franchissement, les jetons sont produits dans les places en sortie en fonction de l'évaluation des arcs

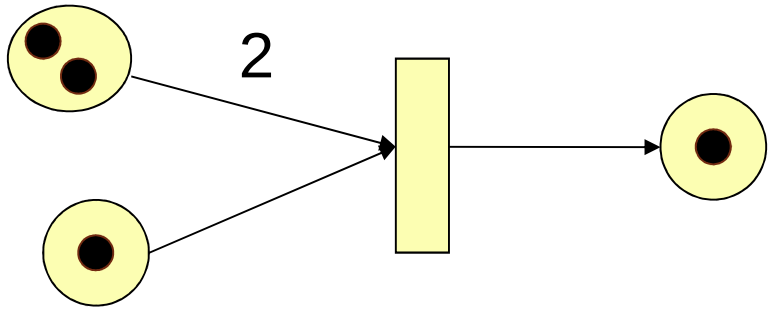
Le nombre de jetons produits et celui des jetons consommés sont indépendants !!

Ressources

- Modélisation des ressources d’un système :
  - Plusieurs jetons
  - Jetons indiscernables
  - Ressources consommées et produites par le franchissement de transitions (événements du système)

Valuation des arcs

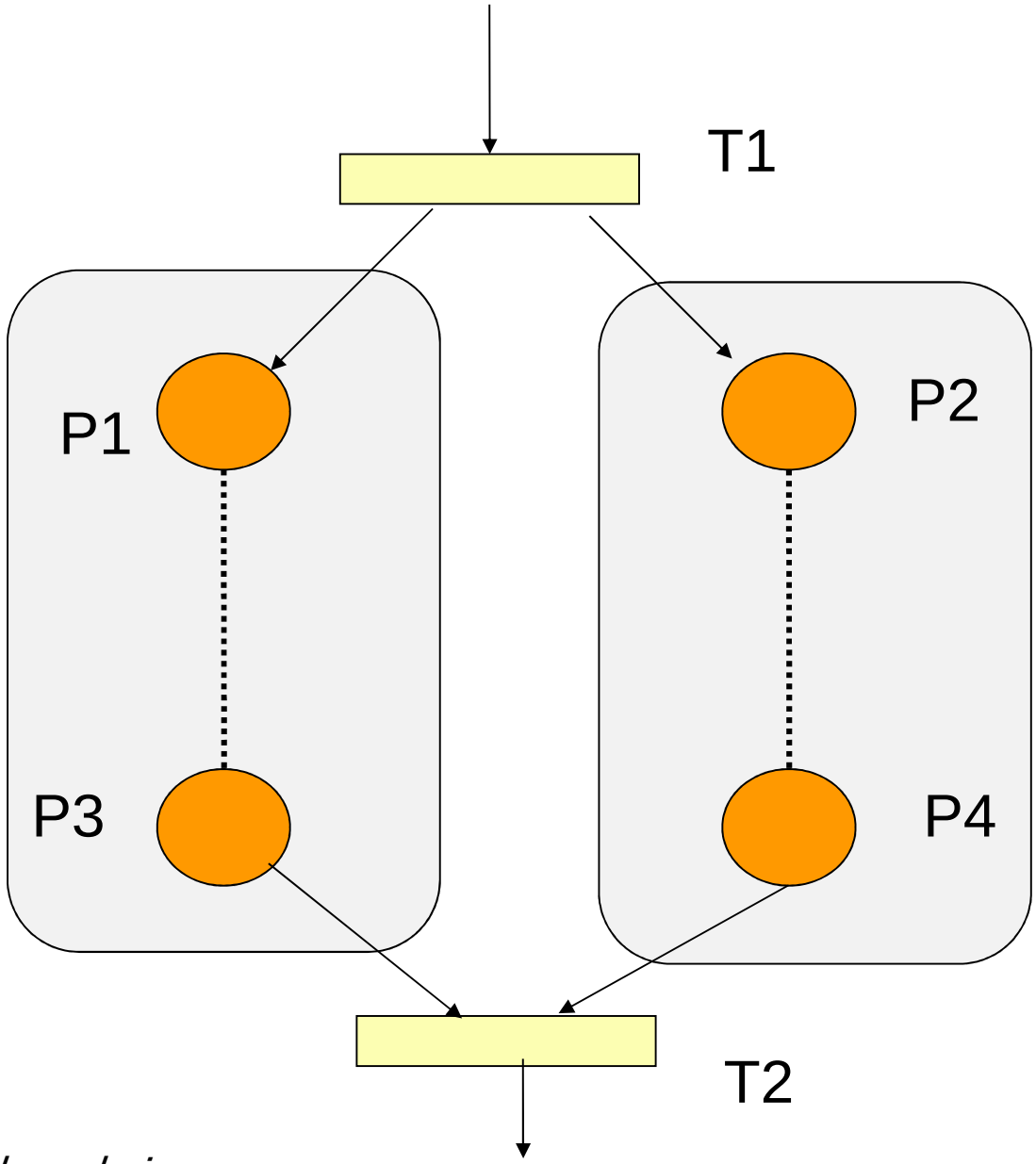
Par défaut, la valuation de l'arc est 1
Nombre de jetons consommés (pré)
Nombre de jetons produits (post)



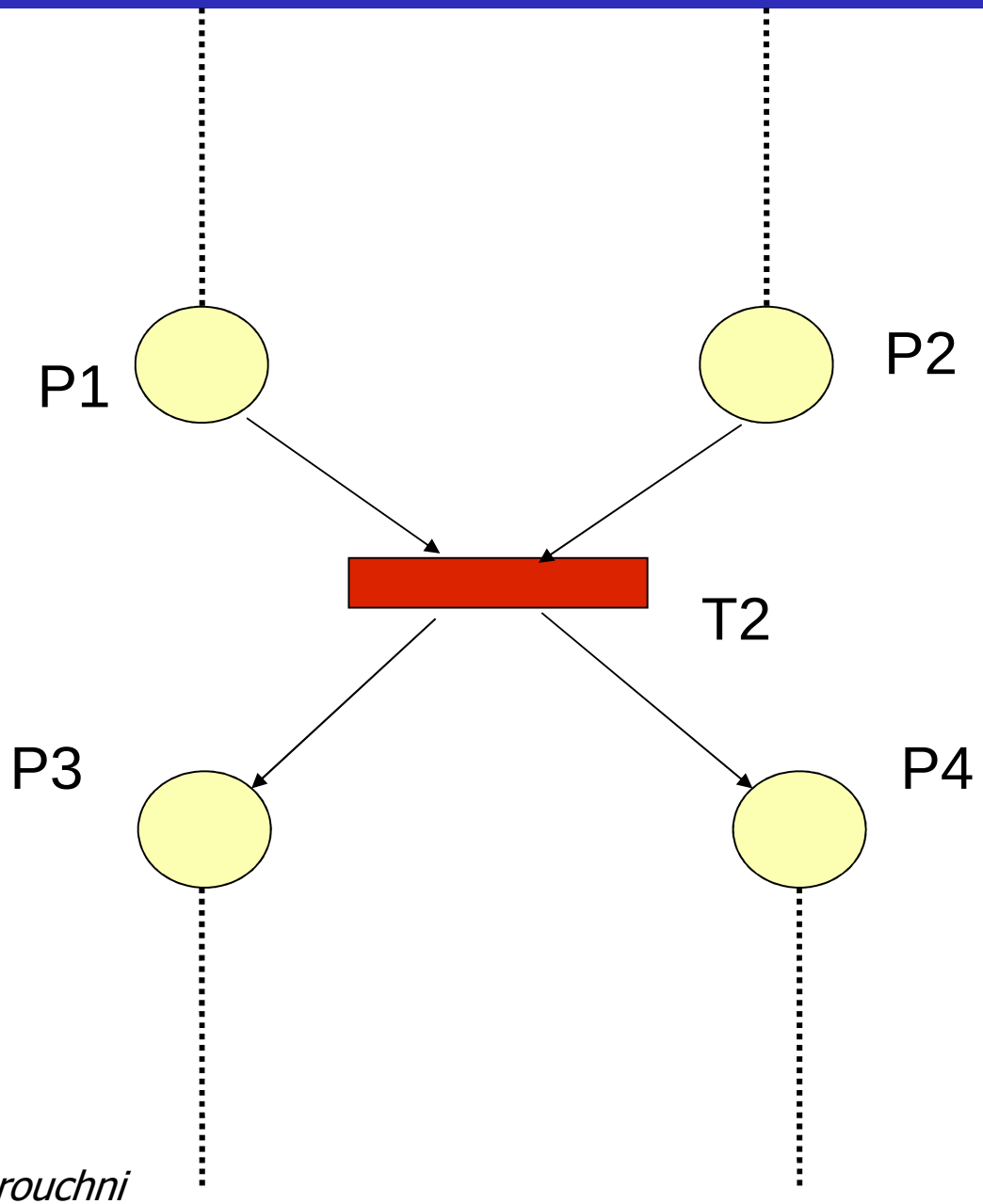
Franchissement

Exemple : 2 cubes et un robot qui va déposer l'un sur l'autre

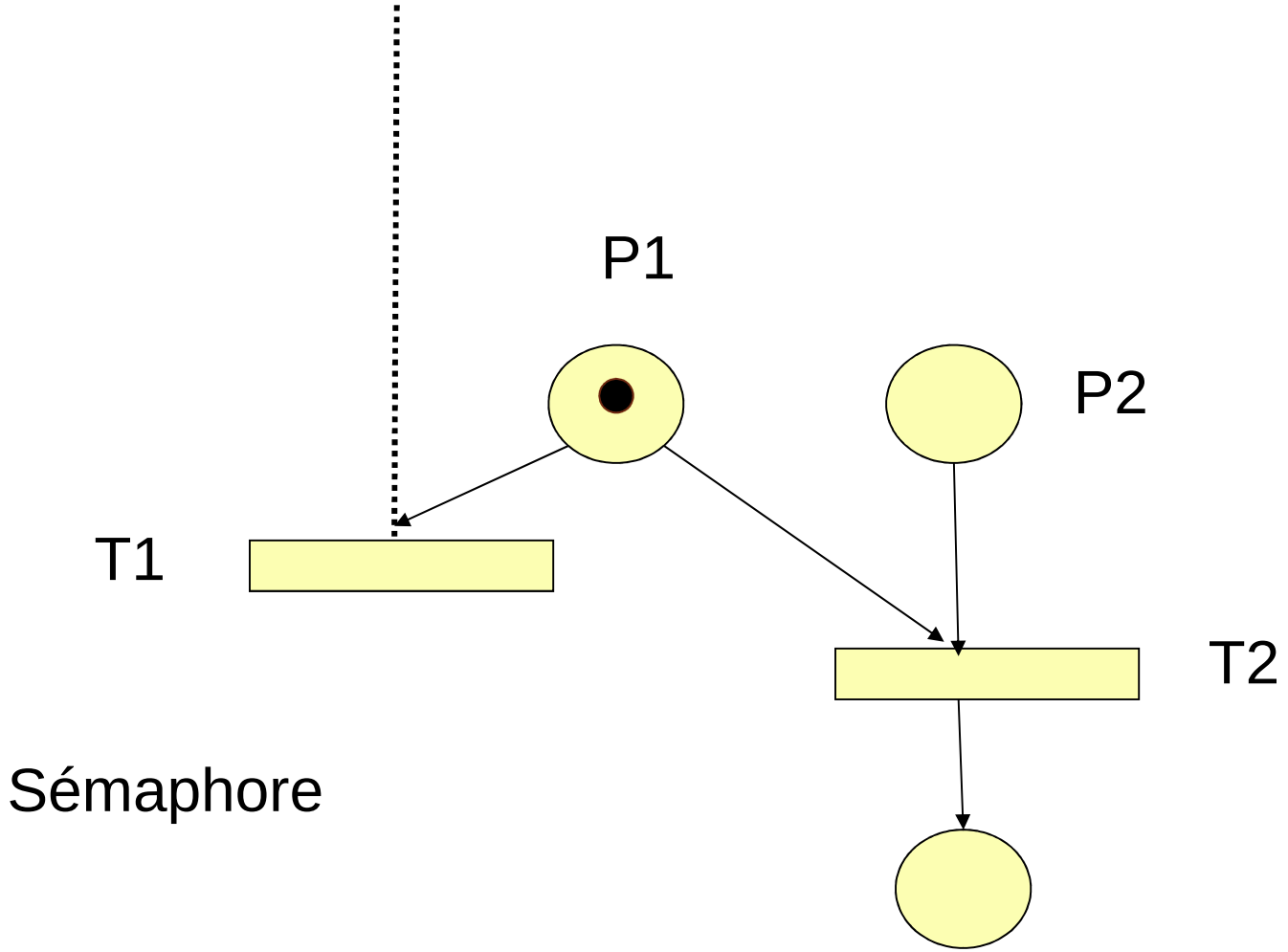
Parallélisme ou concurrence



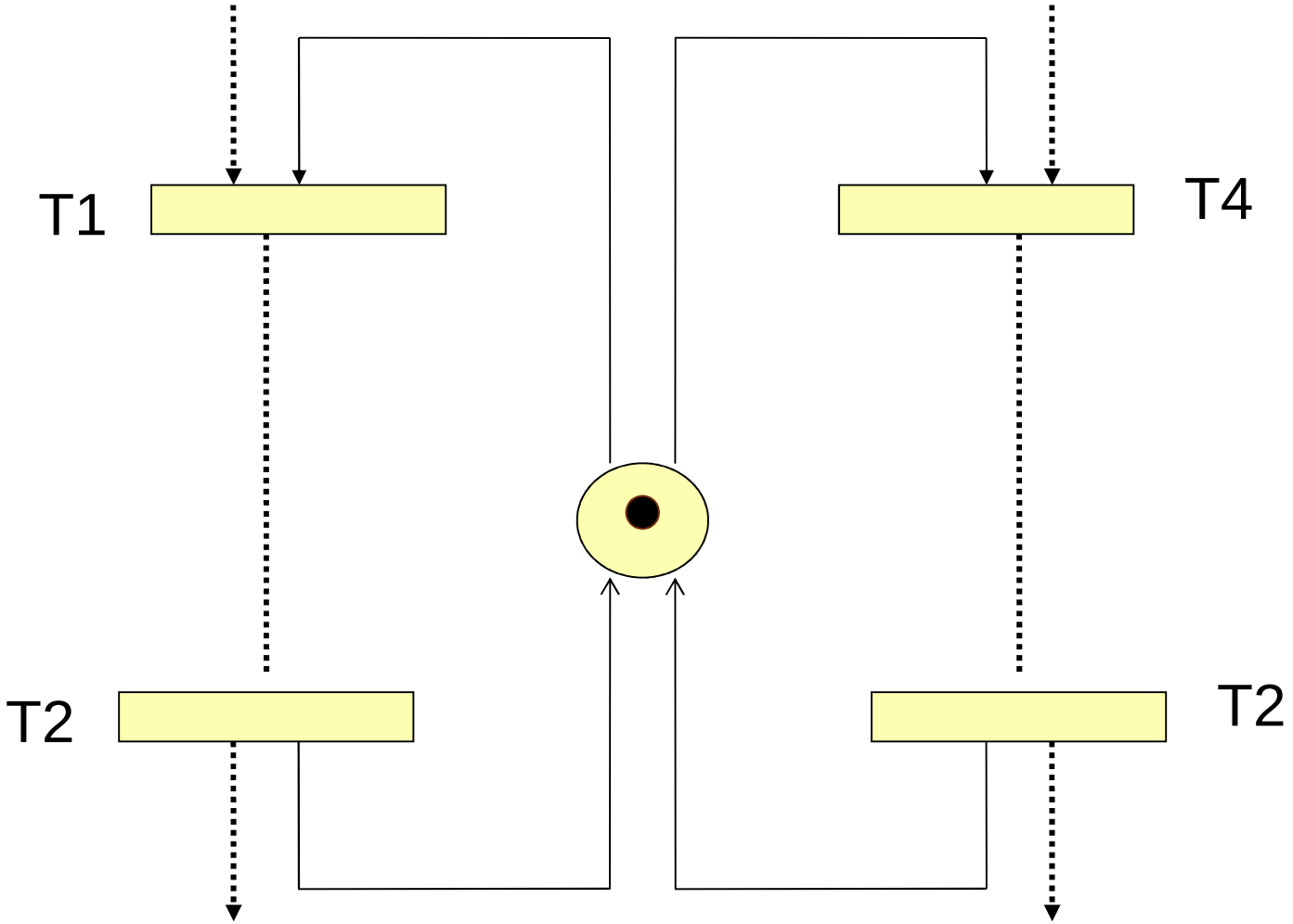
Synchronisation : Rendez-vous



Synchronisation : Sémaphore



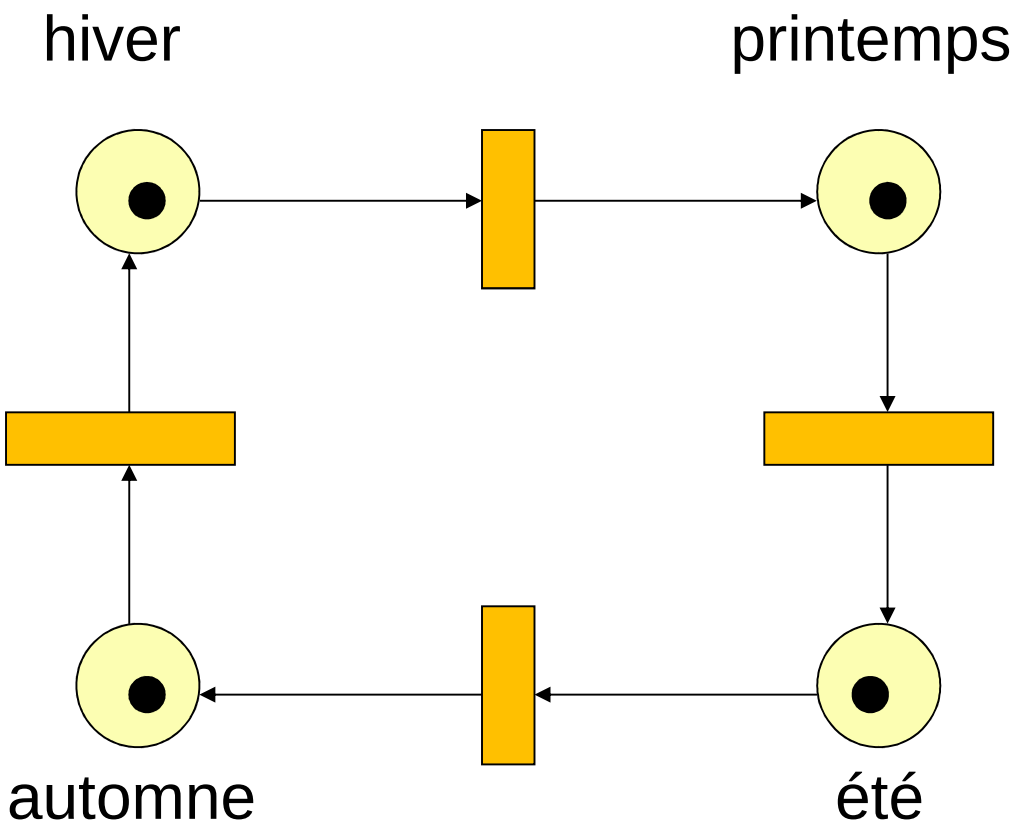
Partage de ressources



Exclusion

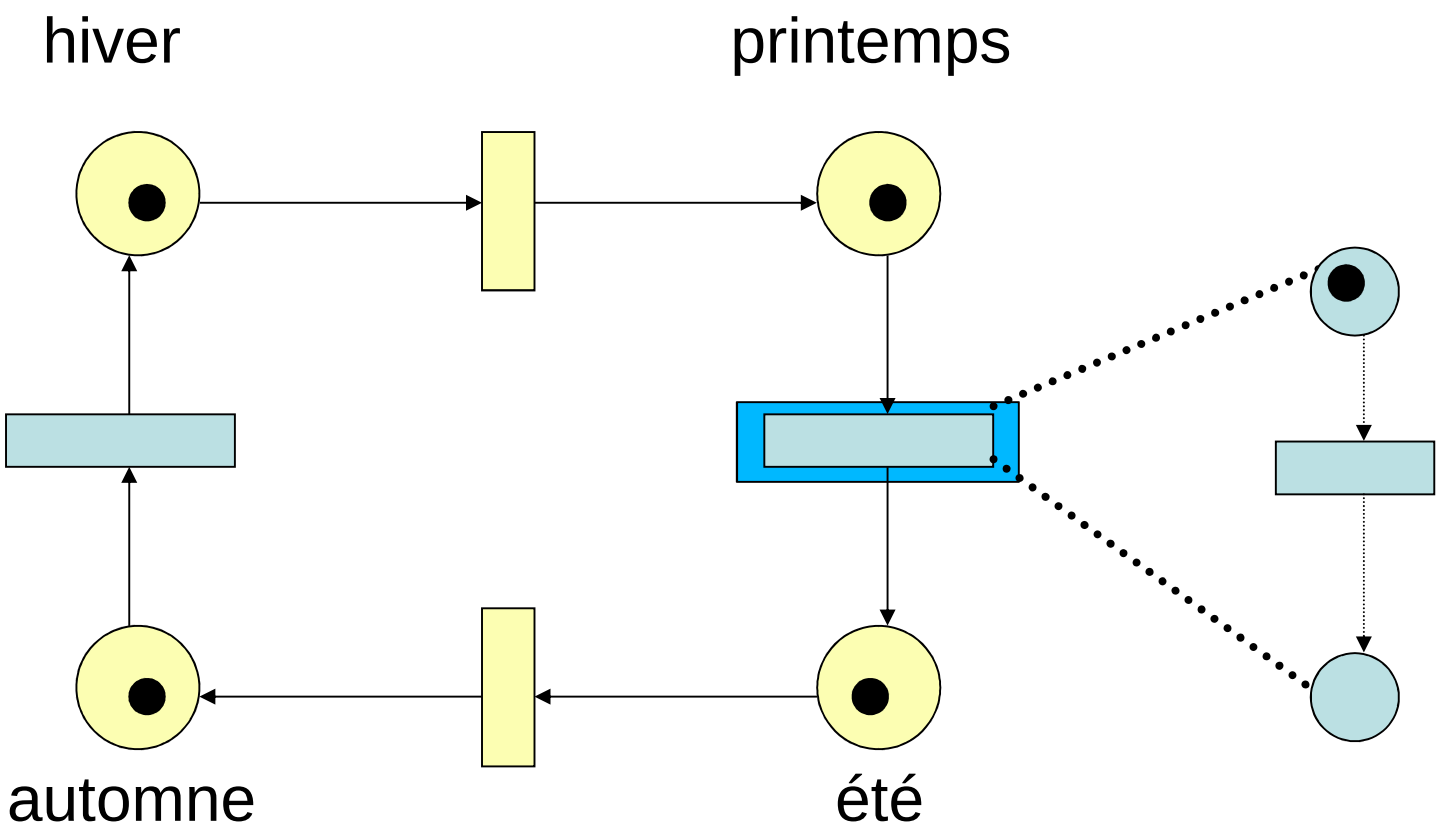
Principales propriétés des RdP

Modèle simulable



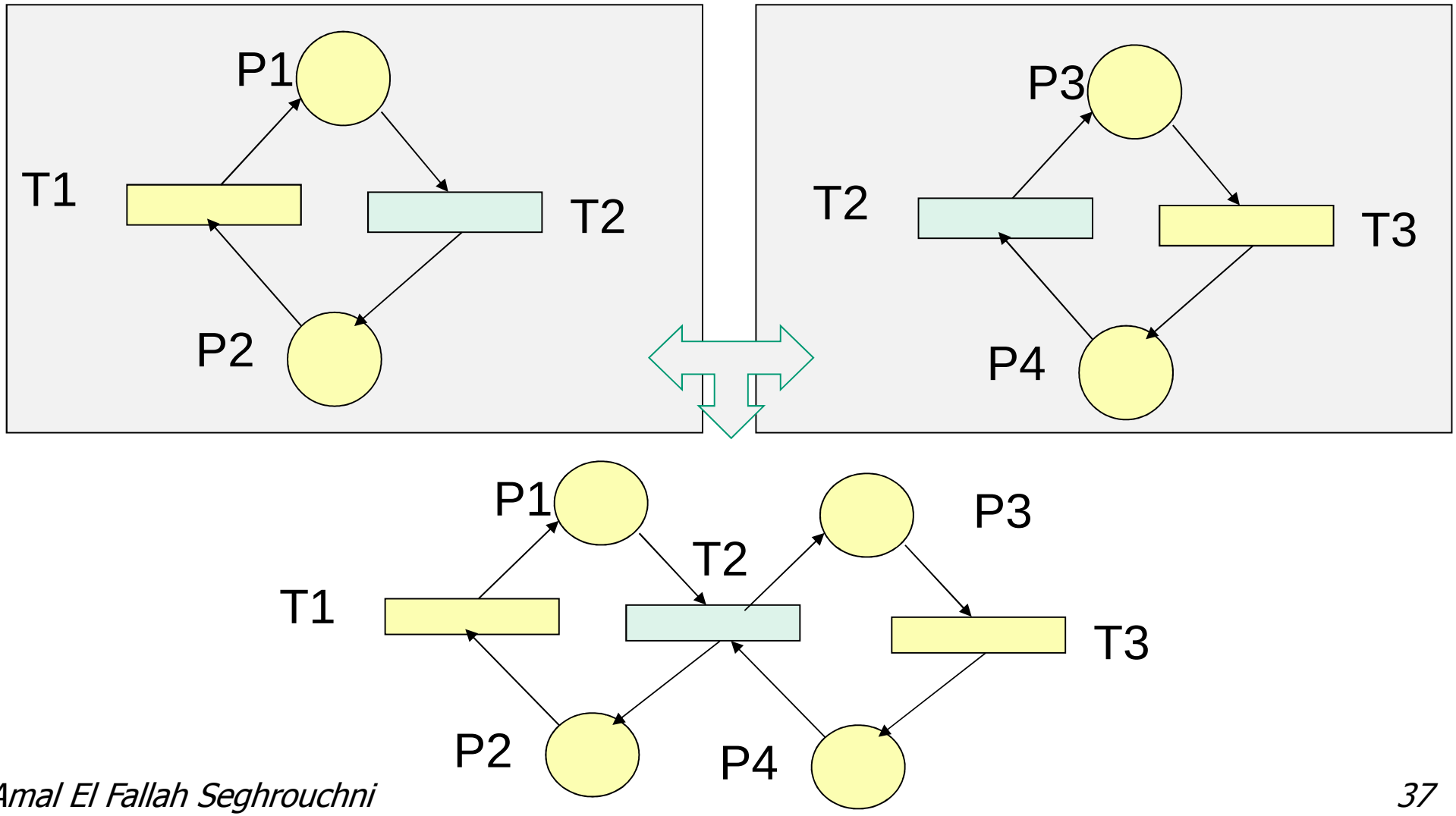
Principales propriétés des RdP

Extensions de ce modèle offrant différents niveaux d'abstraction et des raffinements successifs (ex. Réseaux de Petri Recursifs)



Principales propriétés des RdP

Modèle offrant la composition (ex. transition commune)

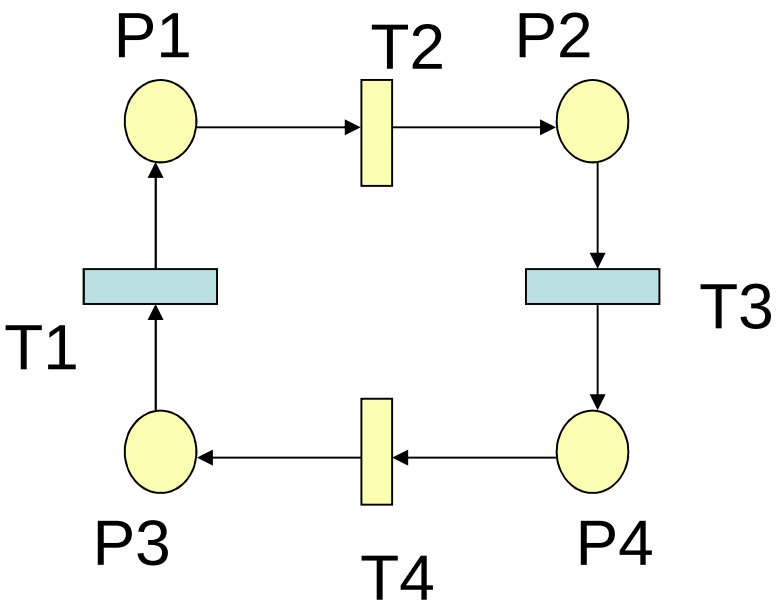


Formalisme

- Un réseau de Petri est un quadruplet R = (P, T, Pré, Post) où :

  - P: ensemble des places
  - T: ensemble des transitions
  - Incidence avant : Pré : PxT $\rightarrow$IN
  - Incidence arrière : Post : PxT $\rightarrow$IN

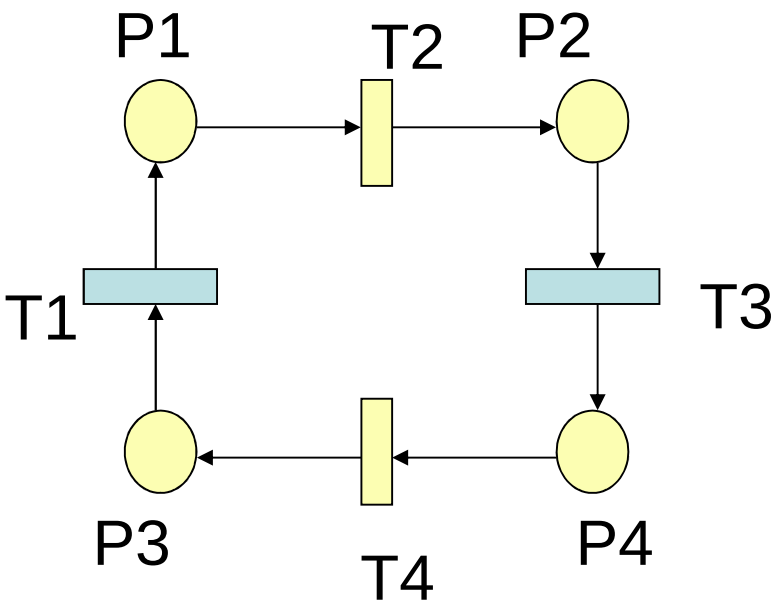
Représentation matricielle matrice d'incidence avant



Valuations des arcs entrants

Pré	T1	T2	T3	T4
P1		1		
P2			1	
P3	1			
P4				1

Représentation matricielle matrice d'incidence arrière

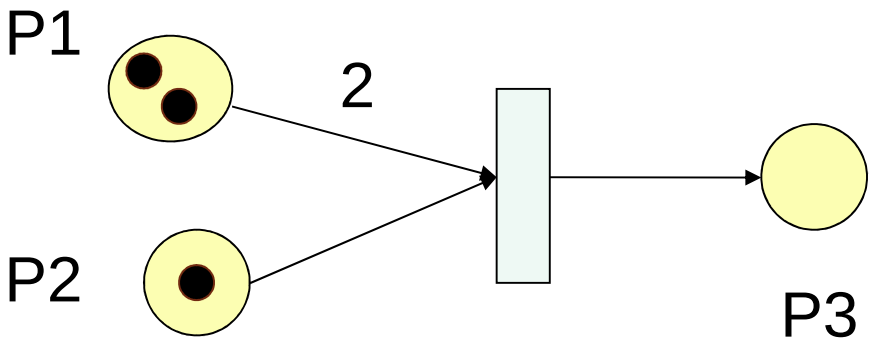


Valuations des arcs sortants

Post	T1	T2	T3	T4
P1	1			
P2		1		
P3				1
P4			1	

Marquage du réseau

- Le marquage d'un RdP est son état $M : P \rightarrow IN$
- Il donne pour chaque place son marquage, *i.e*. le nombre de jetons qu'elle contient.
- Le marquage initial d'un réseau est M_0

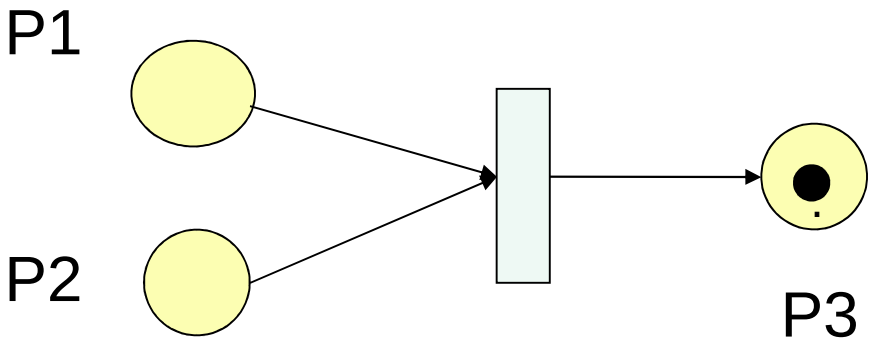


M_0

P1	2
P2	1
P3	0

T

Après le franchissement de T



M_1

P1	0
P2	0
P3	1

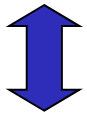
Sémantique

- Une transition t est franchissable ssi :

$$\forall p \in P,\ M(p) \geq Pré\ (p,t)$$

- Le franchissement de t à partir du marquage M produit le marquage M' tel que :

$$M' = M - Pré(.,t) + Post(.,t)$$

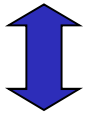


$$\forall p \in P,\ M'(p) = M(p) - Pré(p,t) + Post(p,t)$$

Sémantique

- Marquage M' accessible à partir de M s'il existe une séquence s de transitions (t_1, .., t_n):

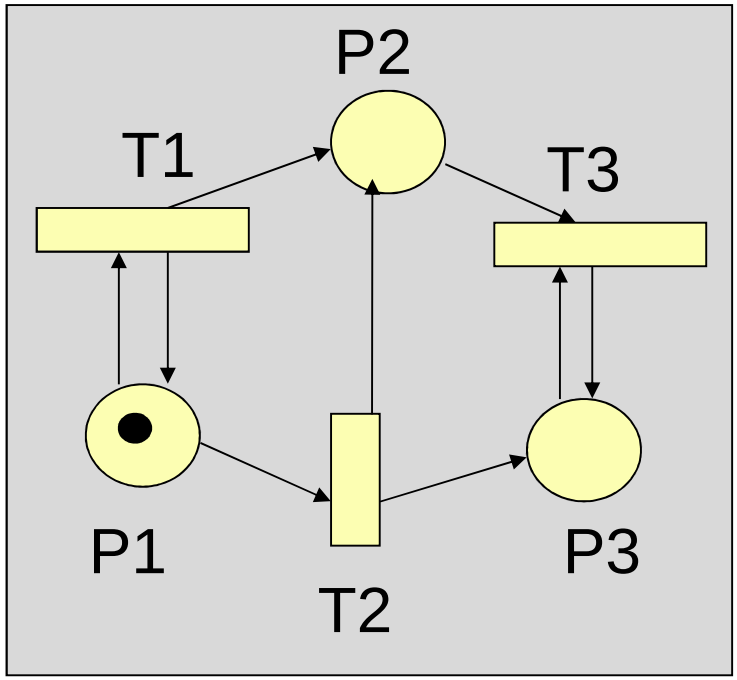
$$M \xrightarrow{t_1} M_1, \qquad M_1 \xrightarrow{t_2} M_2 \qquad M_{n-1} \xrightarrow{t_n} M'$$

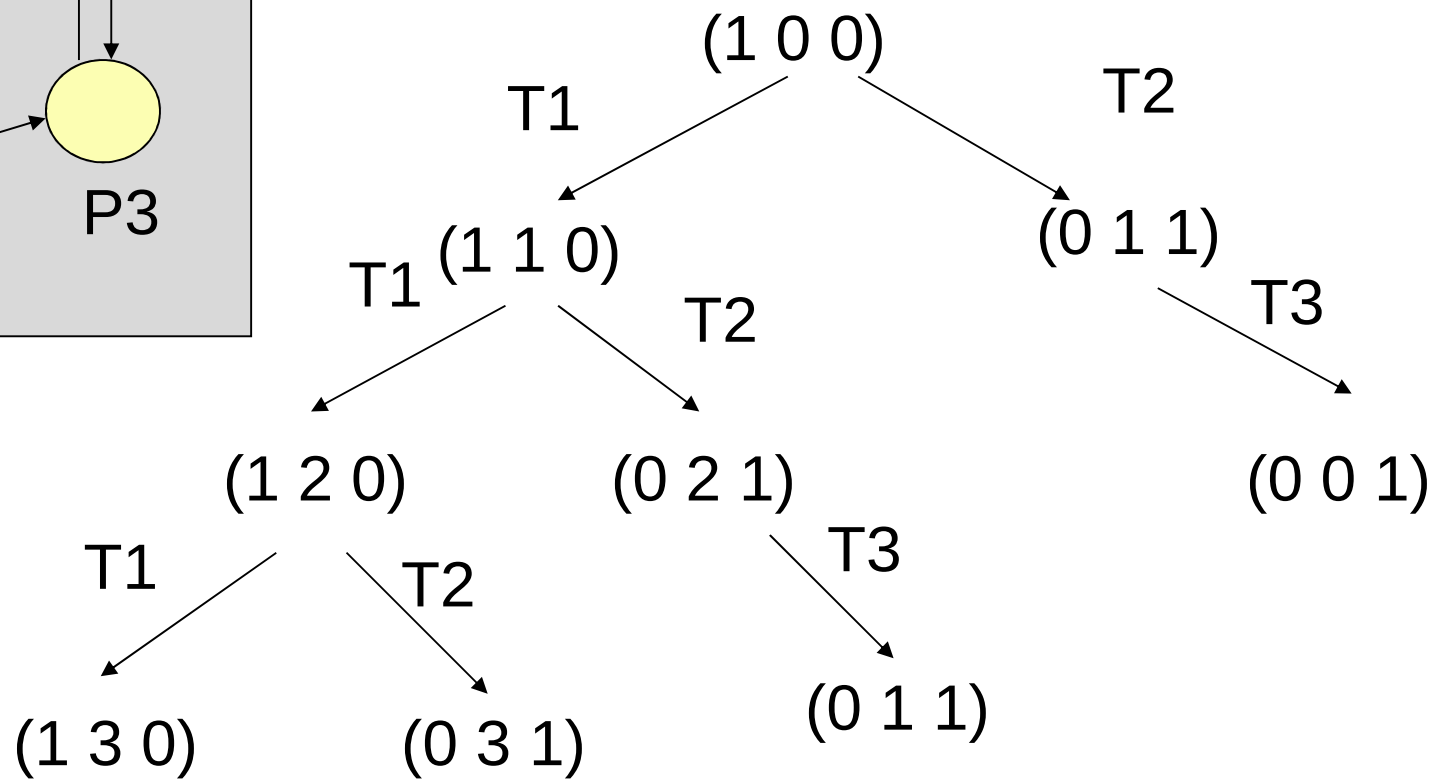


$$M \xrightarrow{s} M'$$

S = (t1, t2, .., tn)

Graphe des marquages accessibles





Propriétés des RdP

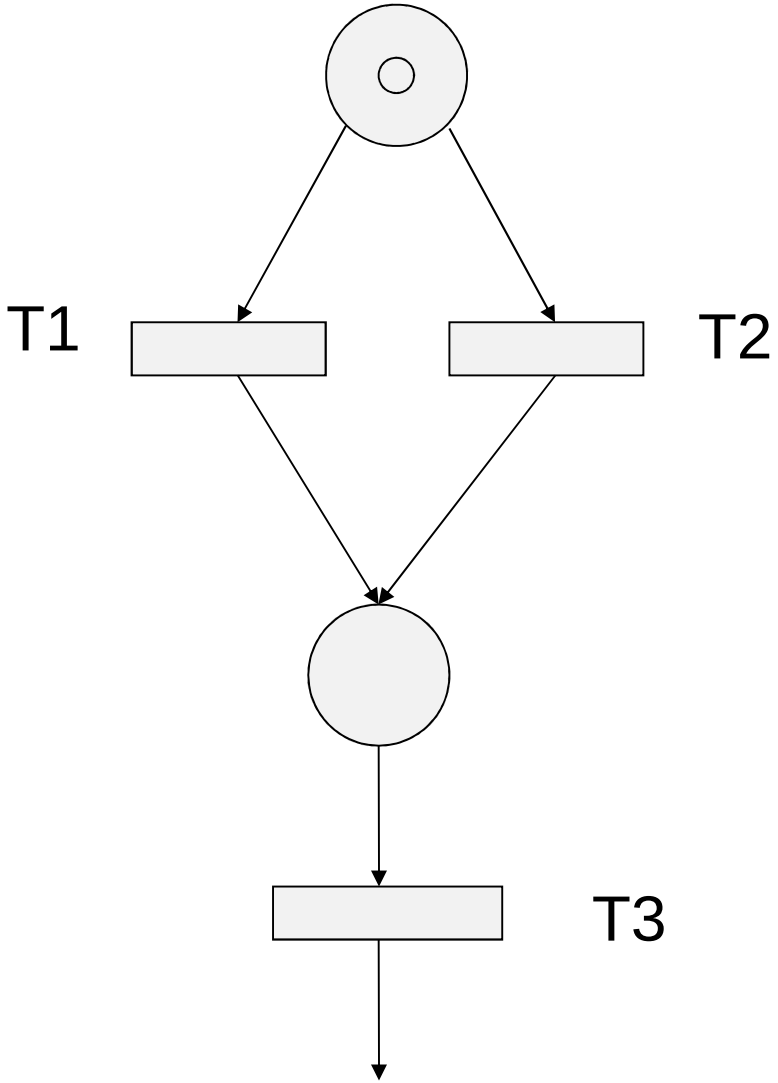
- Propriétés relatives à l'état
  - Le nombre de jetons circulant est-il borné ?

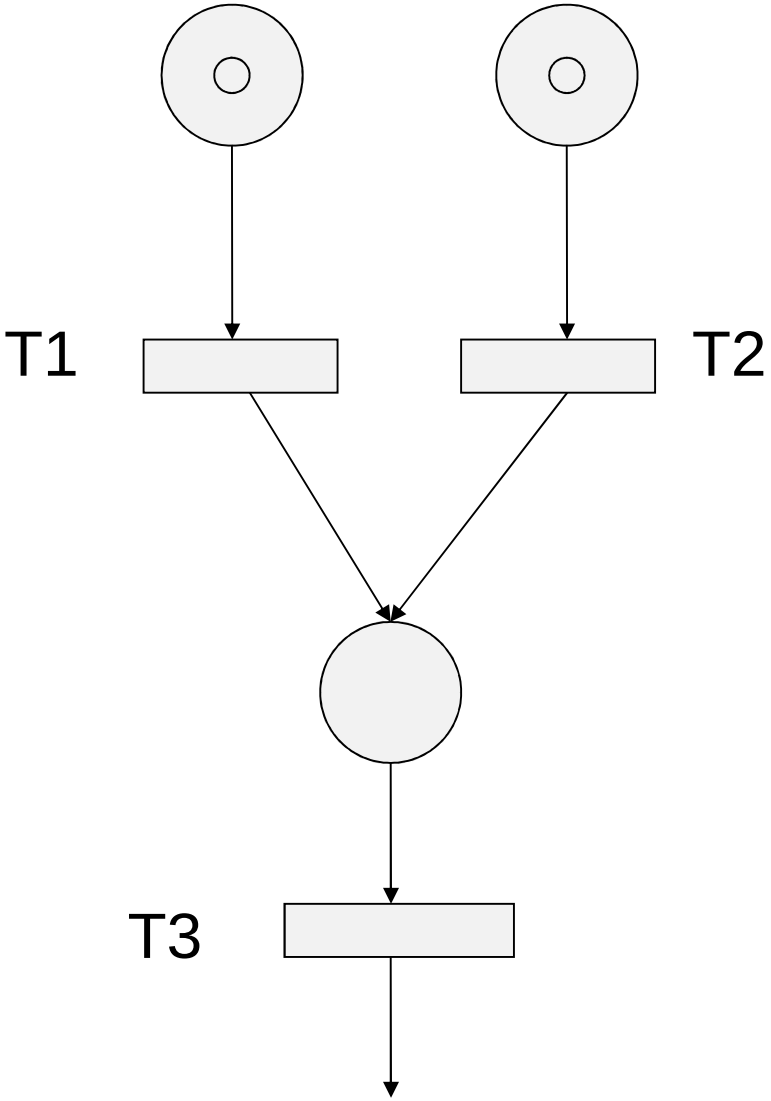
- Un réseau marqué est borné si toutes ses places sont bornées
  - Une place est *k-bornée* si pour tout marquage accessible à partir du marquage initial M0, $M(p) \leq K$

Propriétés relatives à l’activité

- Le réseau ou une partie du RdP peut toujours évoluer ?
  - Un réseau est quasi-vivant si toutes ses transitions le sont
    - Une transition est quasi-vivante si elle peut être franchie au moins une fois (Il existe au moins une évolution du marquage qui permet de la franchir)
  - Un réseau est vivant si toutes ses transitions le sont
    - Une transition est vivante si elle peut être franchie au moins une fois quelque soit l ’évolution du marquage

Exemple

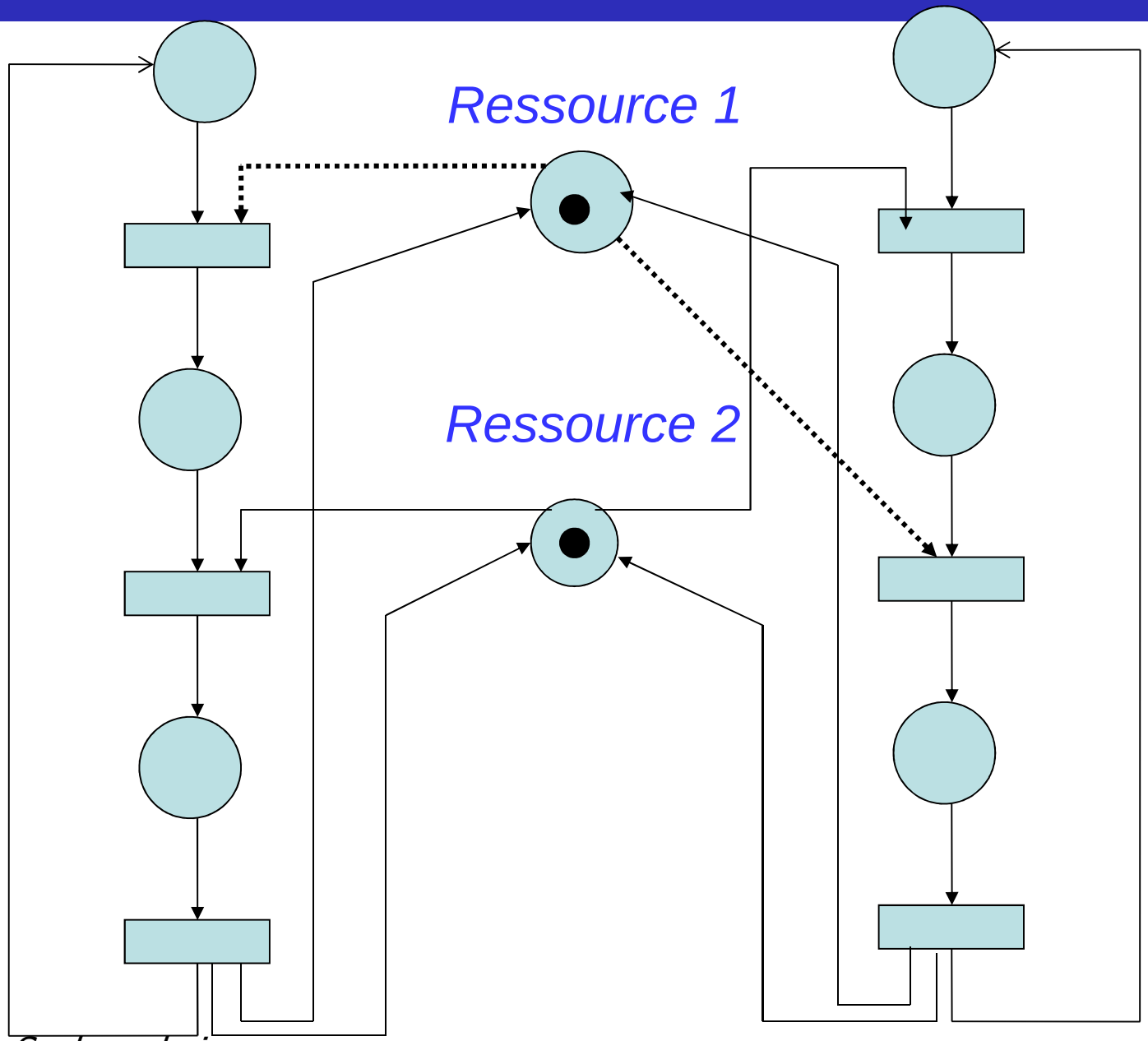




Absence de blocage

- Le réseau a toujours la possibilité d'évoluer *(plus faible que la vivacité)*
- Marquage puits et réseau sans blocage
  - Un marquage puits est un marquage à partir duquel aucune transition n'est franchissable
  - Un réseau marqué est sans blocage si aucun de ses marquages accessibles n'est un marquage puits
- Remarque
  - Un réseau peut être sans blocage même si aucune de ses transitions n'est vivante

Interblocage



Réseaux de Petri

Vérification des propriétés

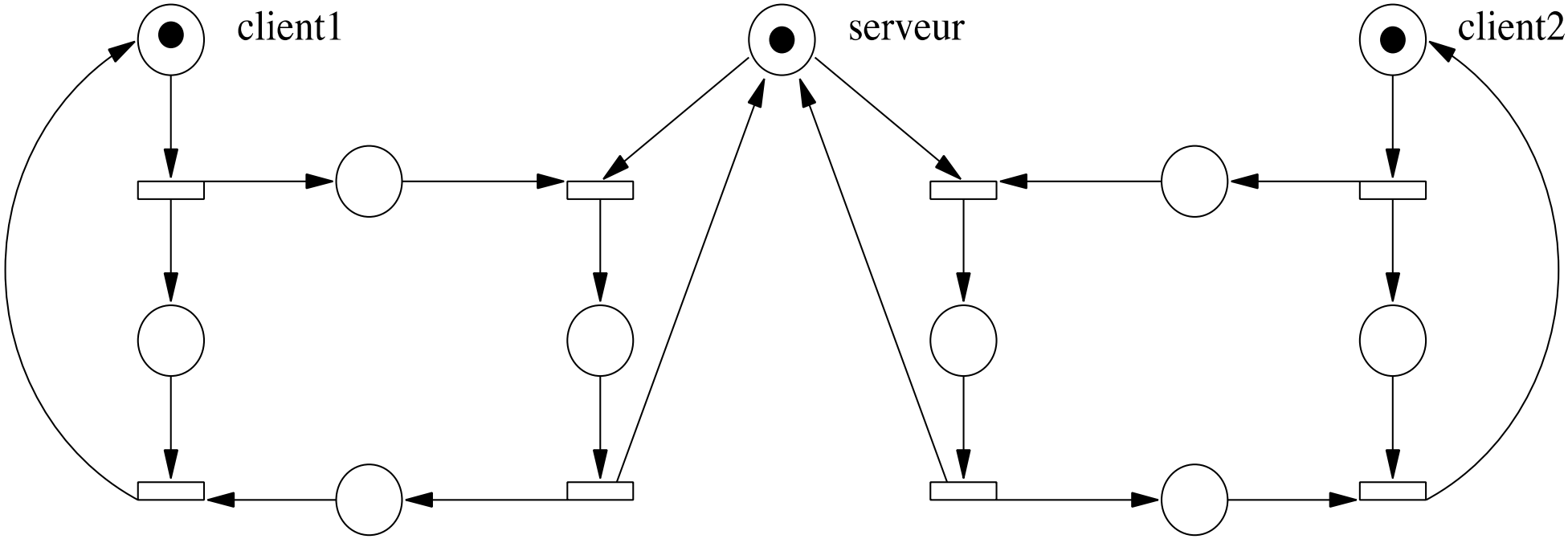
Graphe des marquages

- Construction du GMA

  - Graphe fini : cas idéal car toutes les propriétés peuvent être déduites simplement
  - Graphe infini : les propriétés ne peuvent être déduites : on construit *le graphe de couverture*

Exemple client - serveur

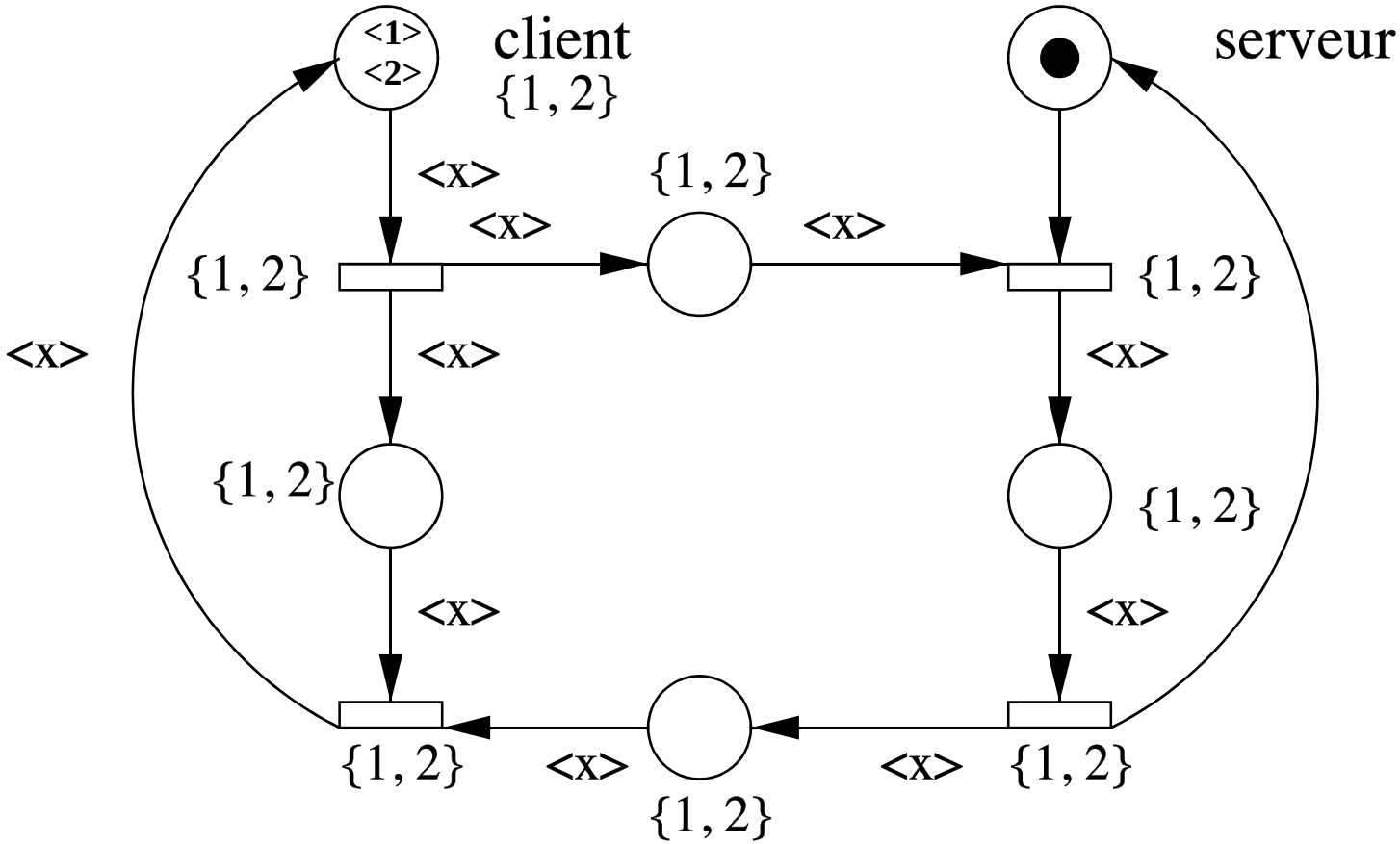
- Deux clients et un serveur



Extension : RdP Colorés

- Toujours 2 clients et un serveur
- Les jetons peuvent être différentiés (<1>, <2> pour les clients, ou « classique » pour le serveur)

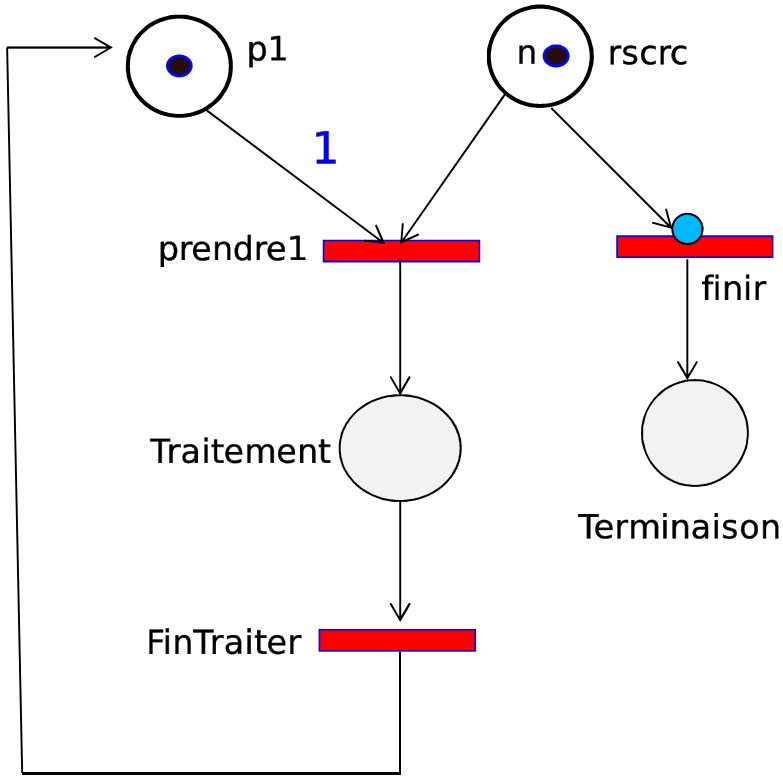
Modèle Coloré du Client Serveur



Arcs inhibiteurs

- Arc inhibiteur
  - Un arc inhibiteur de valuation *n* est validé si la place de départ de l'arc n'a pas n jetons.
  - Cas particulier :

arc de valuation 1, si la place d'entrée est vide de toute marque, alors la transition franchissable



Validation d’un arc inhibiteur

Le franchissement de la transition à laquelle aboutit un arc inhibiteur ne peut se faire que si l'arc inhibiteur est lui même validé.

L'opération de franchissement n'a alors aucune incidence sur le marquage de la place de départ de l'arc inhibiteur. Pour un arc inhibiteur de poids '1', la place qui est vide reste vide.

Les RdP avec arcs inhibiteurs ne se prêtent pas à la validation formelle !!

Autres extensions

- RdP Temporisés
  - Transition franchissable dans un certain intervalle de temps

- RdP Récursifs
  - Les transitions sont typées et la sémantique de leur franchissement dépend du type de transition
    - *Transition élémentaire : franchissement ordinaire*
    - *Transition de fin : franchissement ferme le réseau courant*
    - *Transition abstraite : raffinement par un sous-réseau marqué*