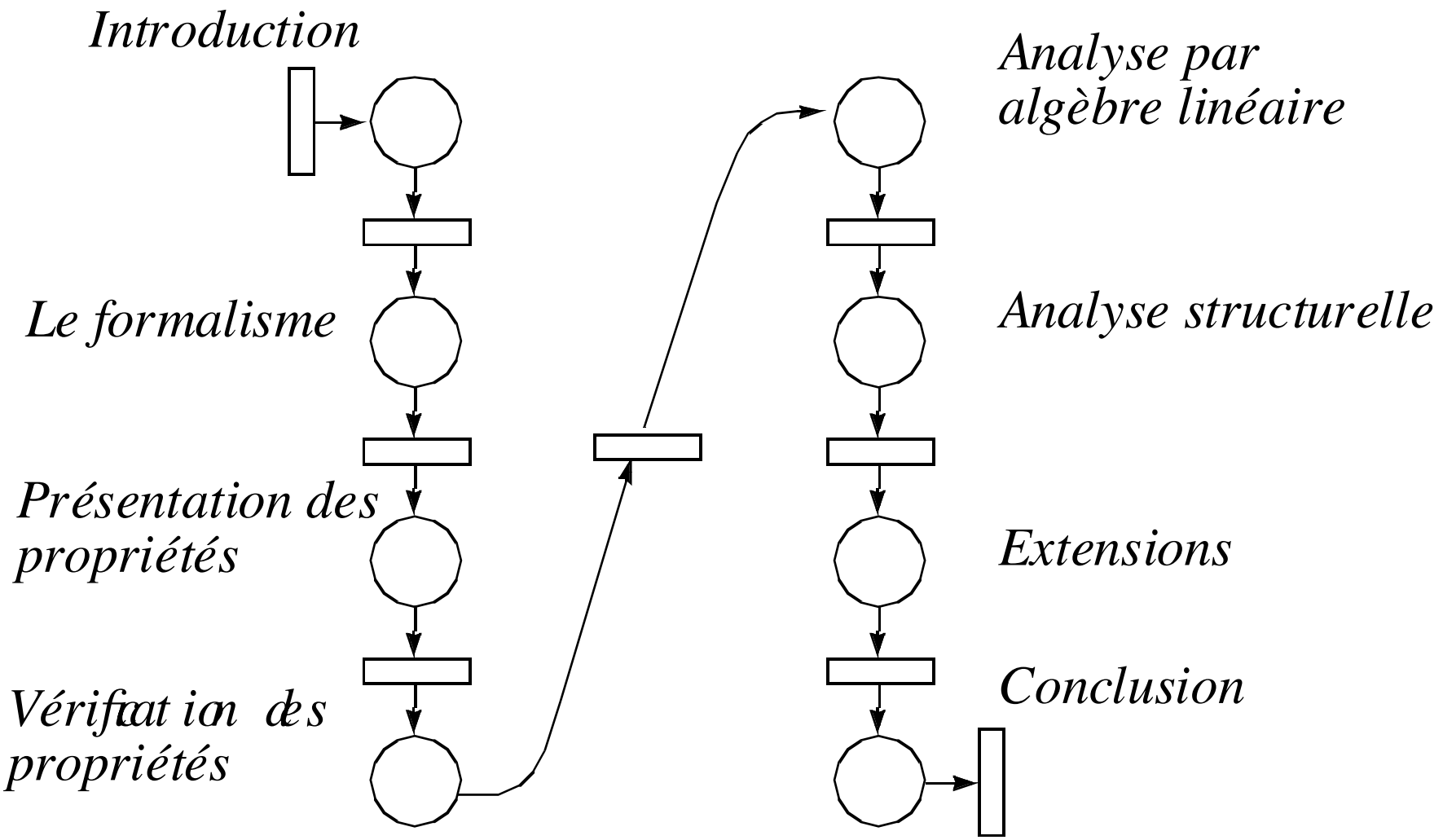
Réseaux de Petri: Introduction

Didier Buchs, Pascal Racloz

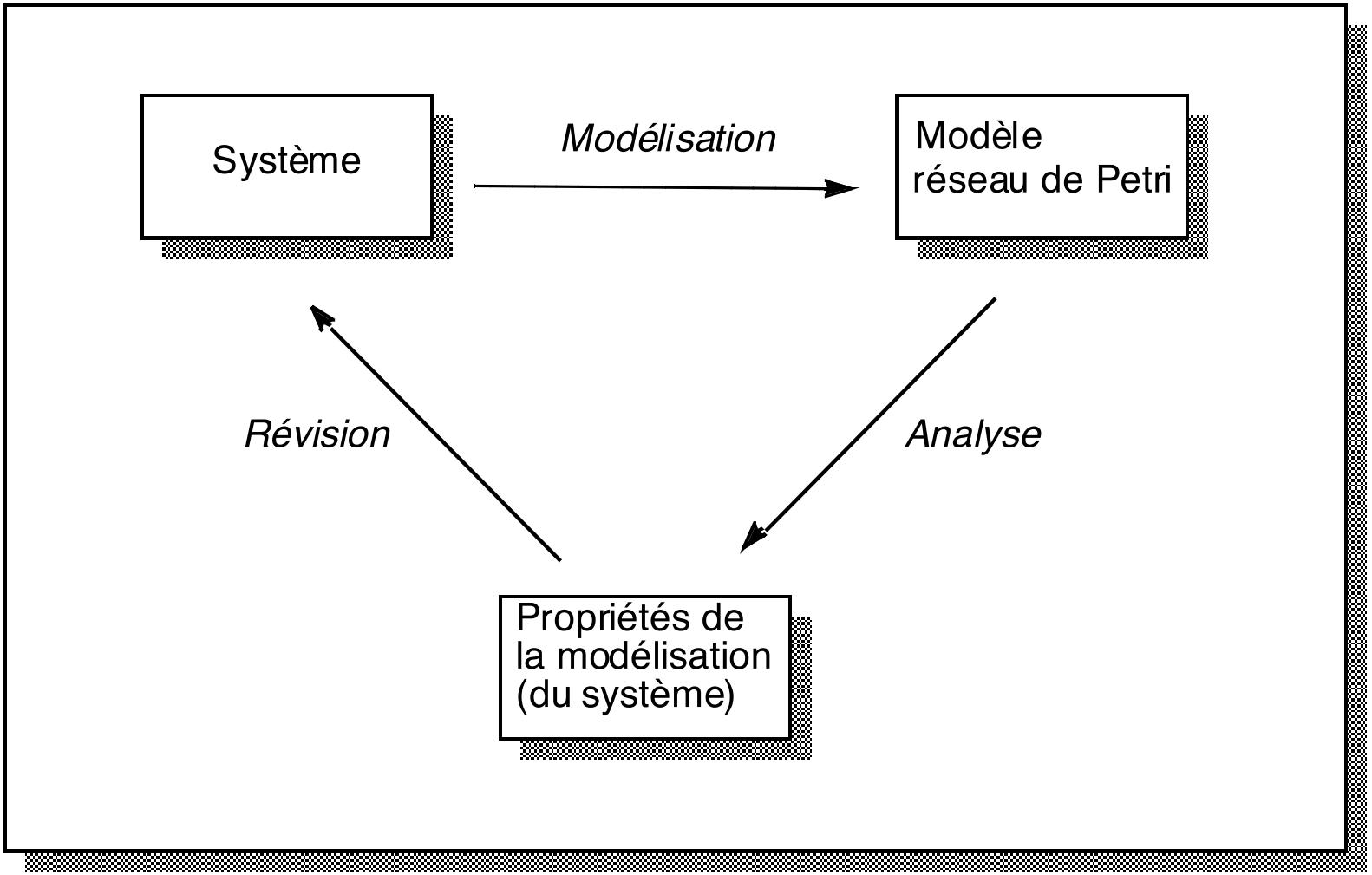
Plan



Introduction

- Motivations
- Eléments de base
- Système ‘conditions-événements’
- Système ‘ressources’
- Quelques modélisations

Motivations



Réseau de Petri (rdP)

- Le formalisme des réseaux de Petri est un outil permettant l'étude de systèmes dynamiques et discrets.
- Il s'agit d'une représentation mathématique permettant la modélisation d'un système.
- L'analyse d'un réseau de Petri peut révéler des caractéristiques importante du système concernant sa structure et son comportement dynamique.
- Les résultats de cette analyse sont utilisés pour évaluer le système et en permettre la modification et/ou l'amélioration le cas échéant.

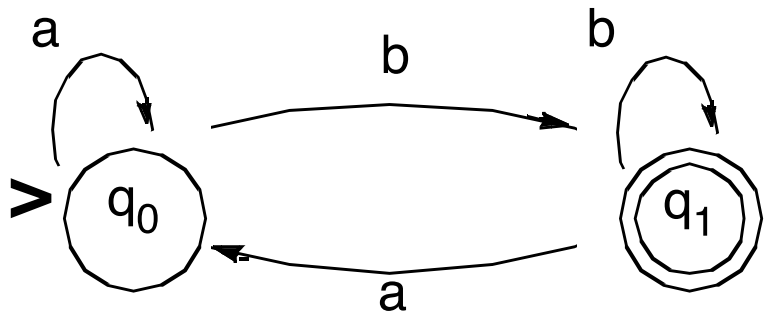
Vocabulaire avec sens intuitif commun

- Modélisation
- Evénement
- Condition

Caractéristiques principales des réseaux de Petri (rdP)

- Distribution des états et des changement d'états dans le réseau
- Dépendance et indépendance d'ensembles d'événements représentées explicitement.
- Représentation à différents niveaux d'abstraction (i.e. détaillés comme abstraits)
- Vérifications des propriétés possibles car basés sur un formalisme mathématique rigoureux
- Modélisation simulable
- Représentation graphique

Automates

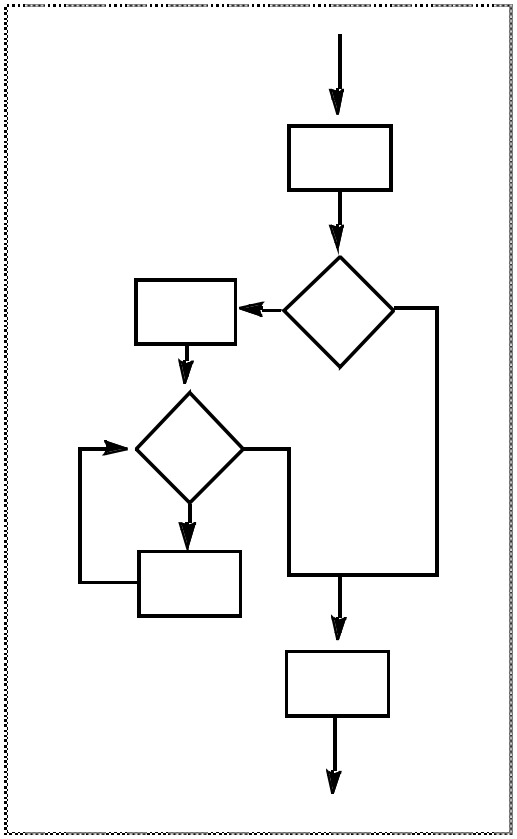


Etat initial: ‘ > ’

Etats accepteurs: ‘ ’

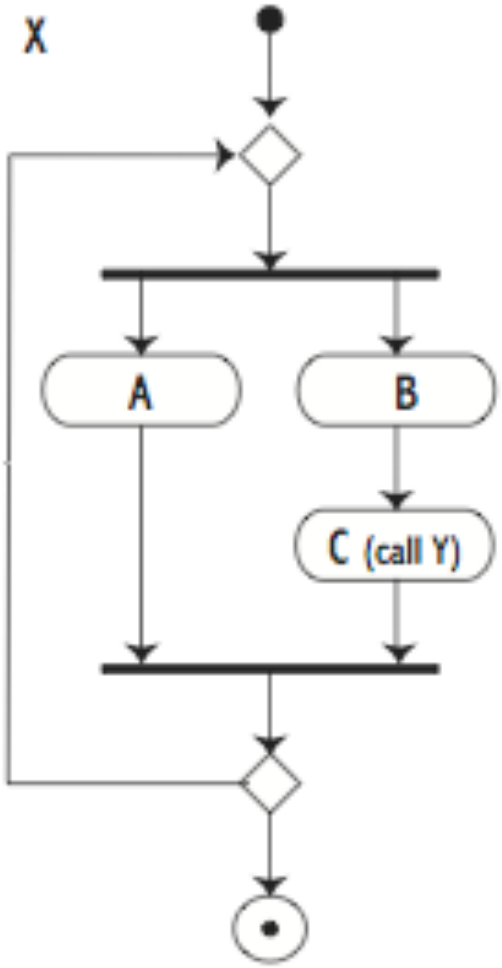
- Automate fini constitué
  - Etats
  - Transitions étiquetées entre ces états
- Permet de décrire un système dont l’état évolue au cours du temps
- Observation et vérification des propriétés

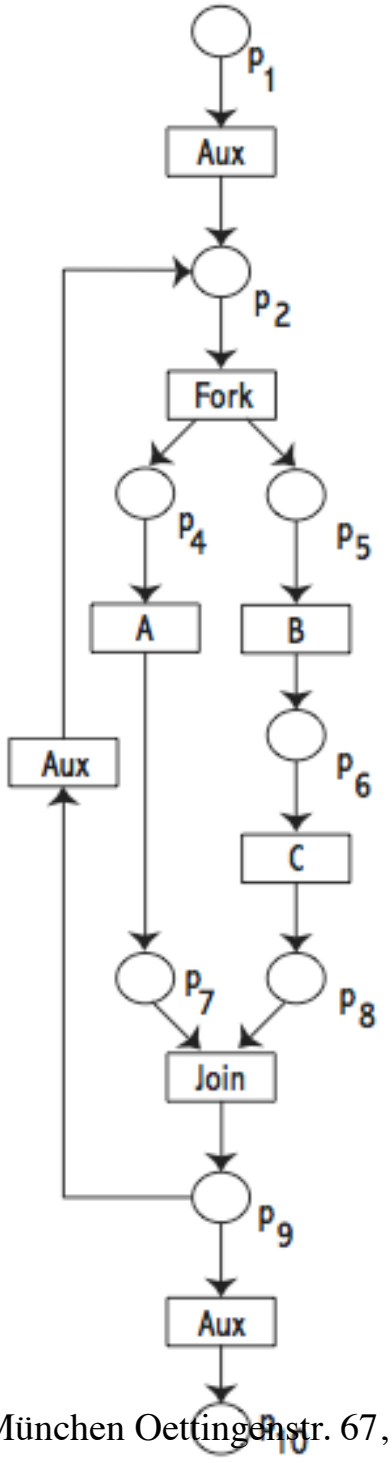
Organigramme (Flowchart)



- Action dépend de l'état (compteur ordinal, valeur de variables,...)
- Son exécution entraîne, le cas échéant, un changement d'état

Activity diagram (UML 2.0)





Semantics of UML 2.0 Activities Harald Störrle Ludwig-Maximilians-Universität München Oettingenstr. 67, 80538 München, GERMANY

Système de transitions

- Notion fondamentale
  - Etats, transitions
  - Pas forcément d'état initial ou d'états terminaux
- Pourquoi ne pas décrire un système à l'aide d'un système de transitions?
  - L'ensemble des états peut être très (trop) grand, voir infini

Modèles de processus

- Les modèles de processus tels que les réseaux de Petri ou les algèbres de processus ne sont pas des systèmes de transitions

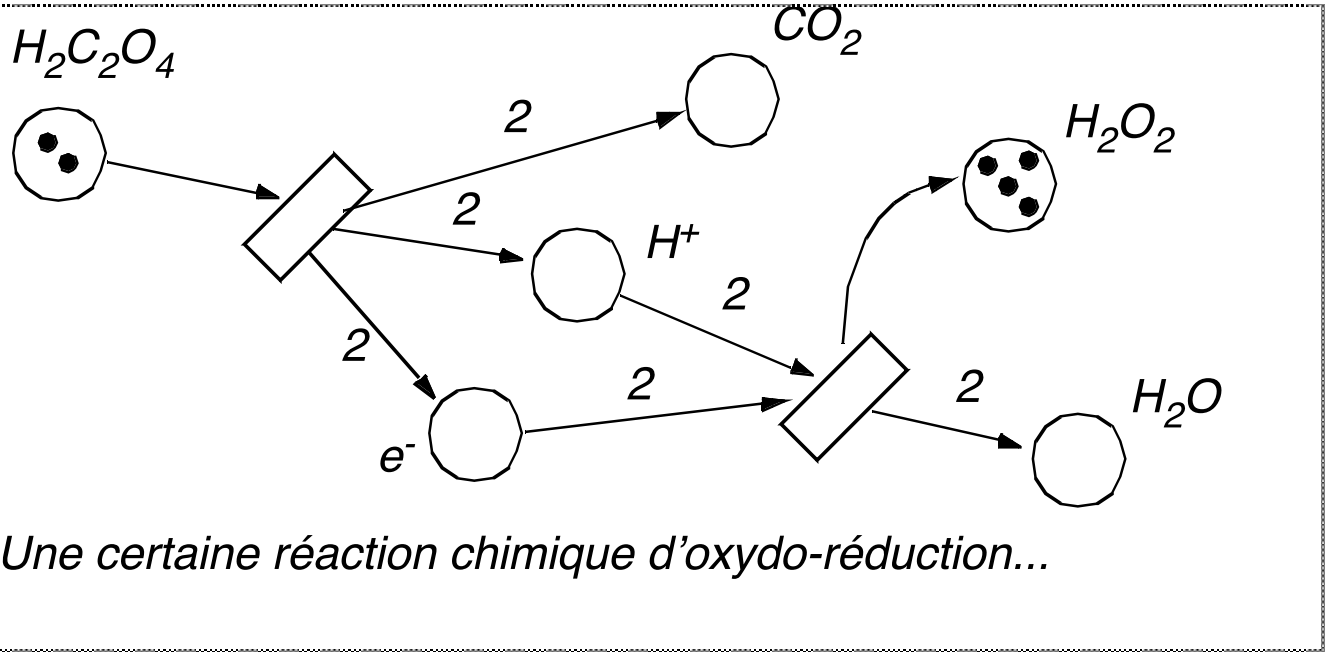
Mais

- il est possible de déduire de la description d'un système dans un de ces formalismes le système de transitions qui représente ses comportements.

Domaines d'application

Problèmes d'organisation, de synchronisation et de coopération entre unités travaillant en parallèle

=> les domaines sont nombreux et variés



Une certaine réaction chimique d'oxydo-réduction...

Concepts de base

- Evénement
  - Les événements sont des actions se déroulant dans le système.
  - Le déclenchement d'un événement dépend de l'état du système.
  - Un état du système peut être décrit comme un ensemble de conditions.
- Condition
  - Une condition est un prédicat ou une description logique d'un état du système.
  - Une condition est vraie ou fausse.
- Déclenchement, précondition, postcondition
  - Les conditions nécessaires au déclenchement d'un événement sont les *préconditions* de l'événement.
  - Lorsqu'un événement se produit, certaines de ses pré-conditions peuvent cesser d'être vraies alors que d'autres conditions, appelées *postconditions* de l'événement doivent devenir vraies.

Exemple:
Atelier de coupe de bois

- Conditions (ou états du système)
  - La machine de coupe est au repos *(c1)*
  - *Une commande est en attente (c2)*
  - *La commande est en cours de traitement (c3)*
  - *La commande est terminée (c4)*
- Evénements
  - Une commande arrive *(e1)*
  - *La machine débute la commande (e2)*
  - *La machine termine la commande (e3)*
  - *La commande est envoyée pour la livraison (e4)*

Evénement	Précondition	Postcondition
e1		
e2		
e3		
e4		

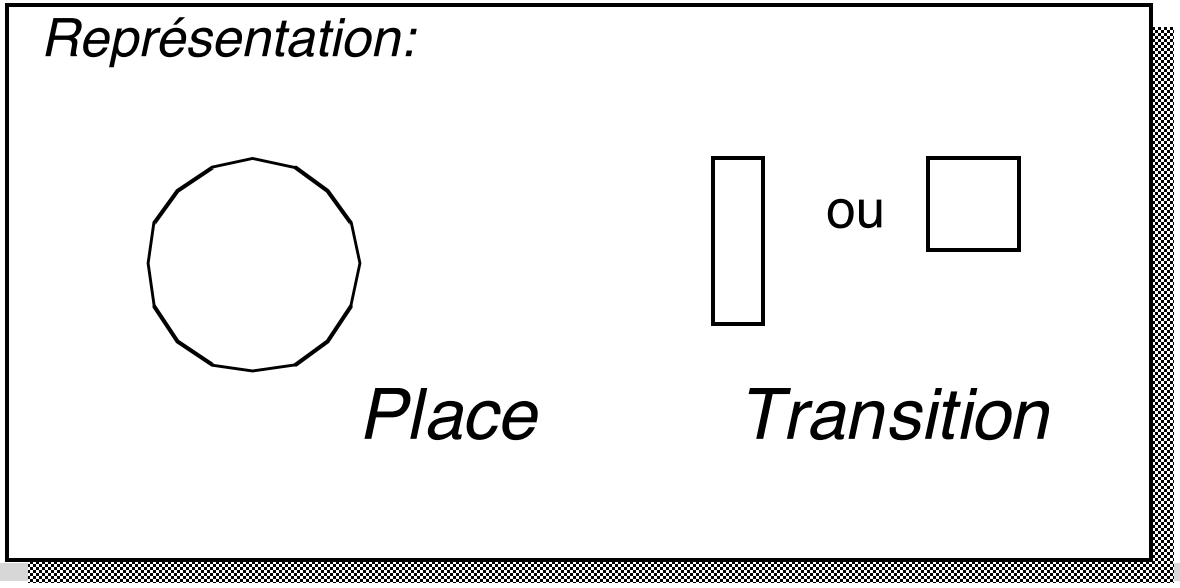
Modélisation d'un système à *événement*

- Condition

modélisée à l'aide d'une PLACE.

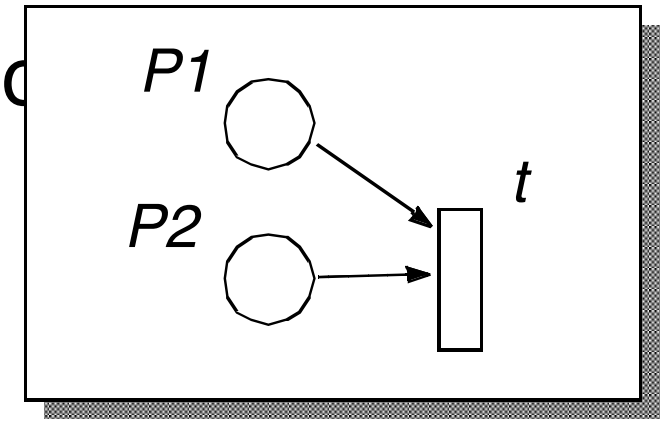
- Evénement

modélisé à l'aide d'une TRANSITION.

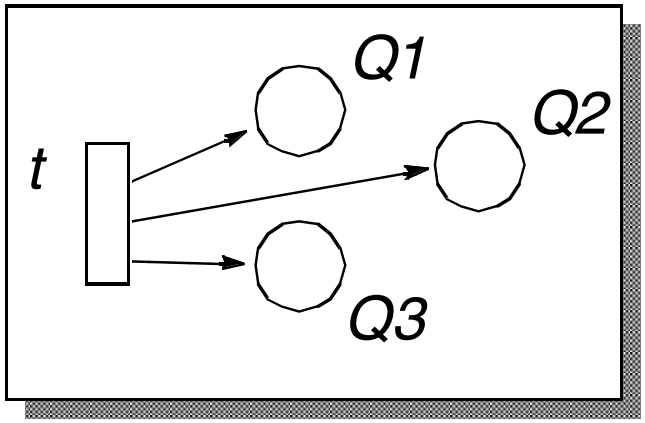


(Cont'd)

- Précondition d'une transitio

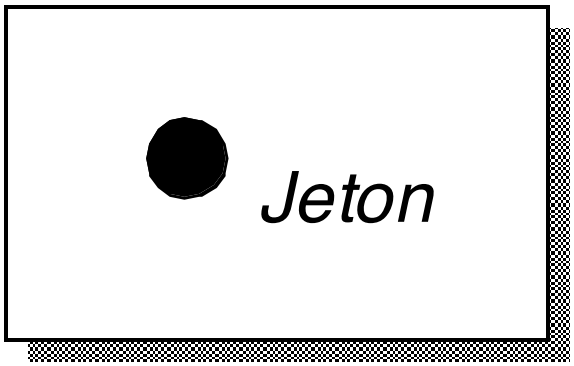


- Postcondition d'une transition



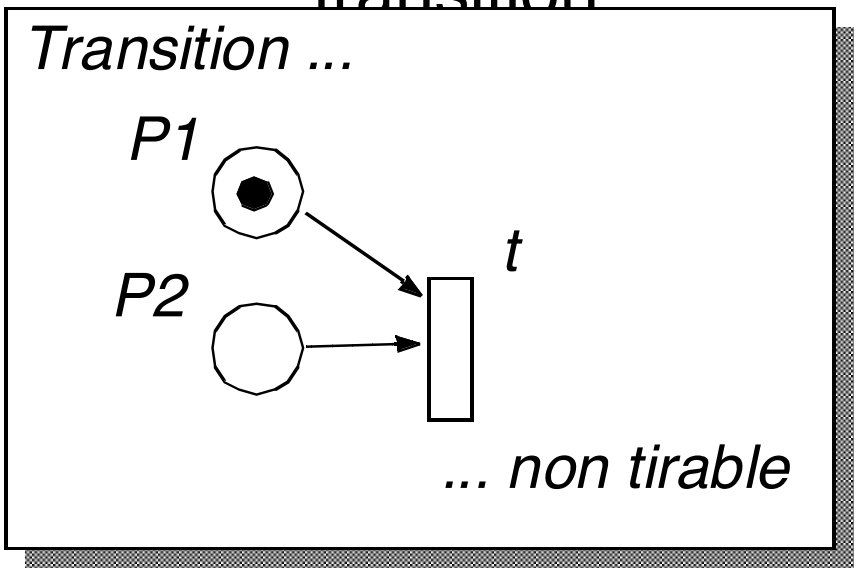
(Cont'd)

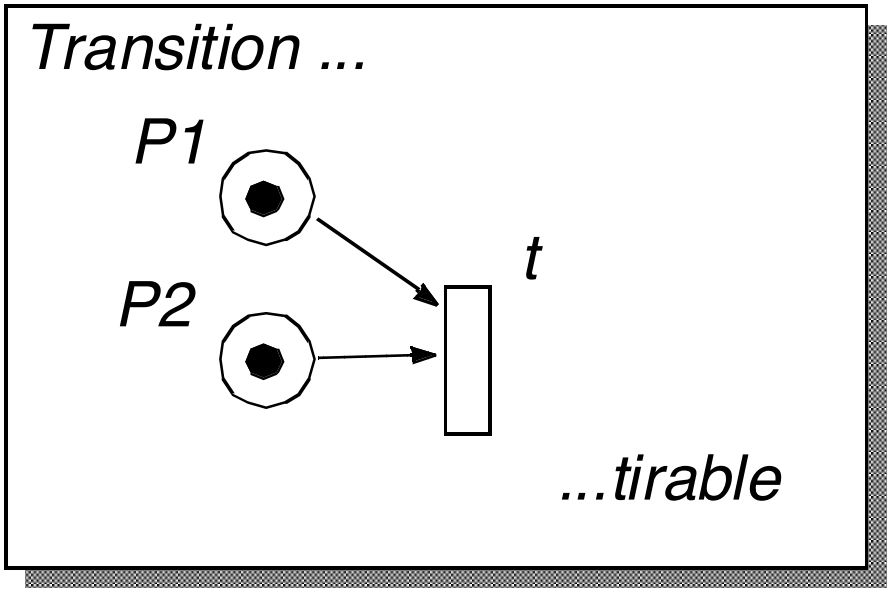
- Satisfaction d'une condition modélisée à l'aide d'un JETON.



(Cont'd)

- Pre-condition de déclenchement d'une transition



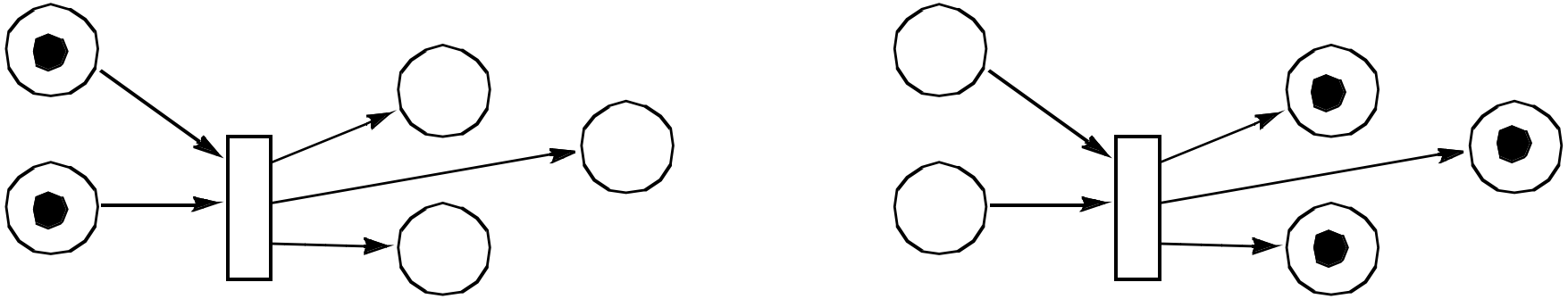


(Cont'd)

- Déclenchement d'une transition tirable

On parle plus volontiers du *tir* ou du *franchissement* d'une transition

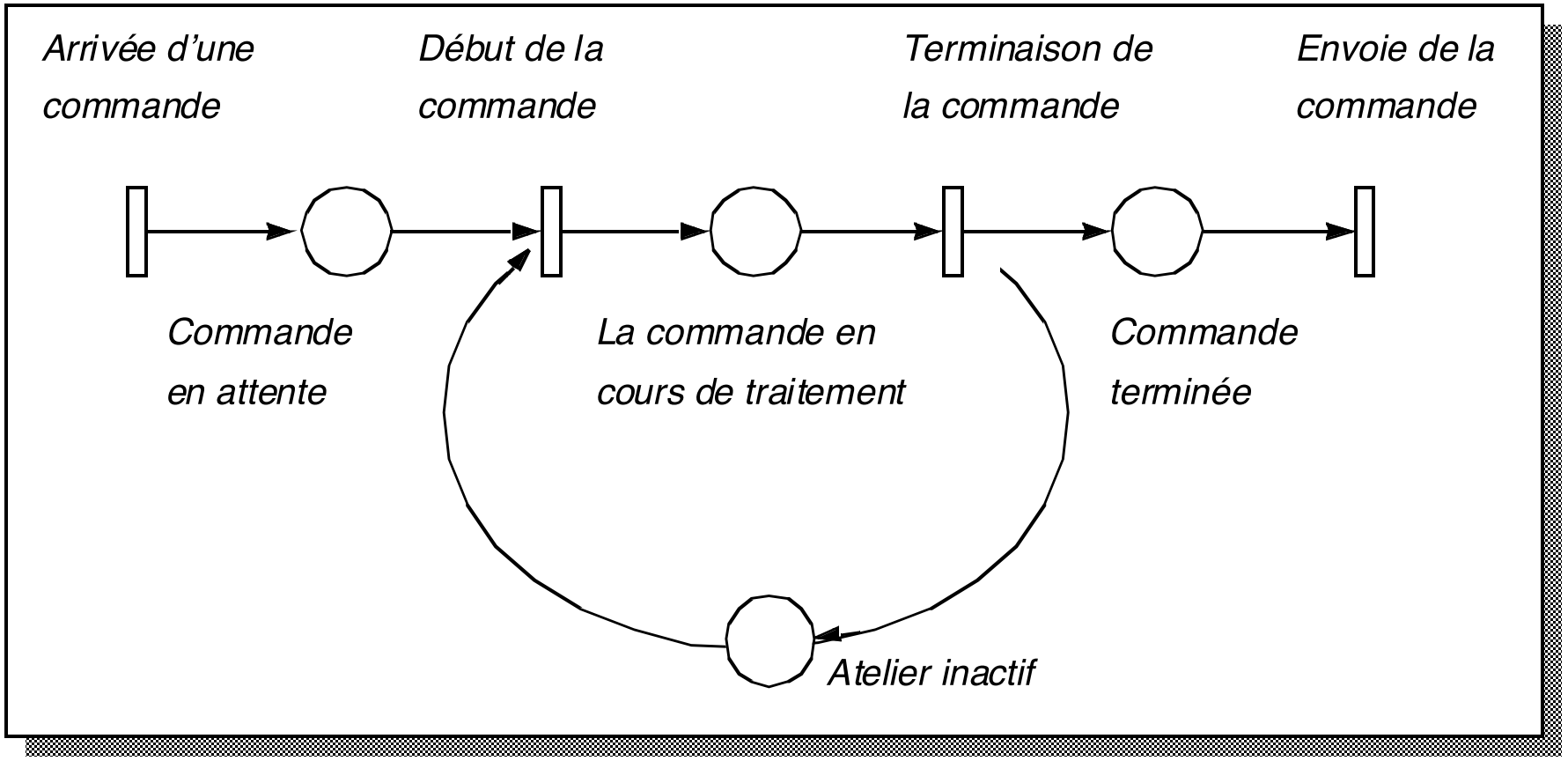
Tir d'une transition



Situation avant...

Situation après...

Exemple: L'atelier de coupe de bois



Exercices

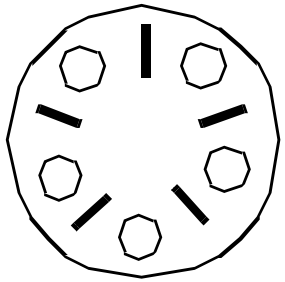
Producteur-consommateur

Un producteur produit des biens, qui sont stockés et ensuite consommés.

Exercices

Les philosophes

Les philosophes pensent ou mangent. Pour pouvoir manger, un philosophe doit saisir ses fourchettes gauche et droite. Lorsqu'il a terminé, il libère ses fourchettes. Il y a 5 philosophes. Indiquer différentes modélisations possibles selon la façon de saisir les fourchettes.



Exercices (...)

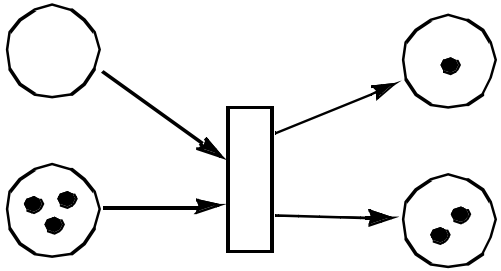
Modélisation de systèmes avec *ressources*

- Pour certains systèmes, il est plus juste de raisonner en termes d'ensemble de ressources, au sens large, qu'en termes de conditions-événements.
- Places: elles peuvent contenir plusieurs jetons.
- Le nombre de jetons contenus dans une place reflète le nombre de ressources qu'elle possède.
- Les jetons d'une place n'ont pas d'identité individuelle, autrement dit ils sont indiscernables.
- Ces ressources sont consommées et produites par les événements du système.

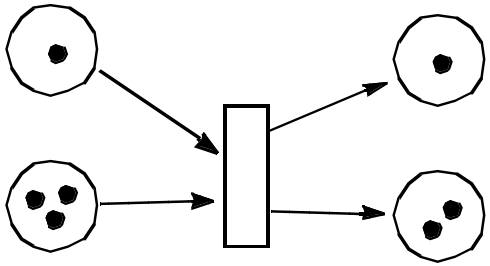
(Cont'd)

- Tir d'une transition

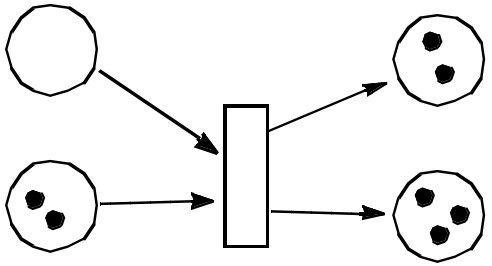
Transition... non tirable



Transition tirable ...

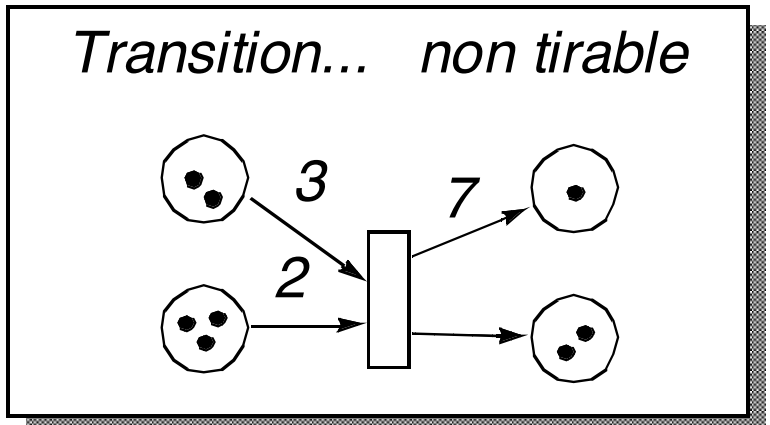


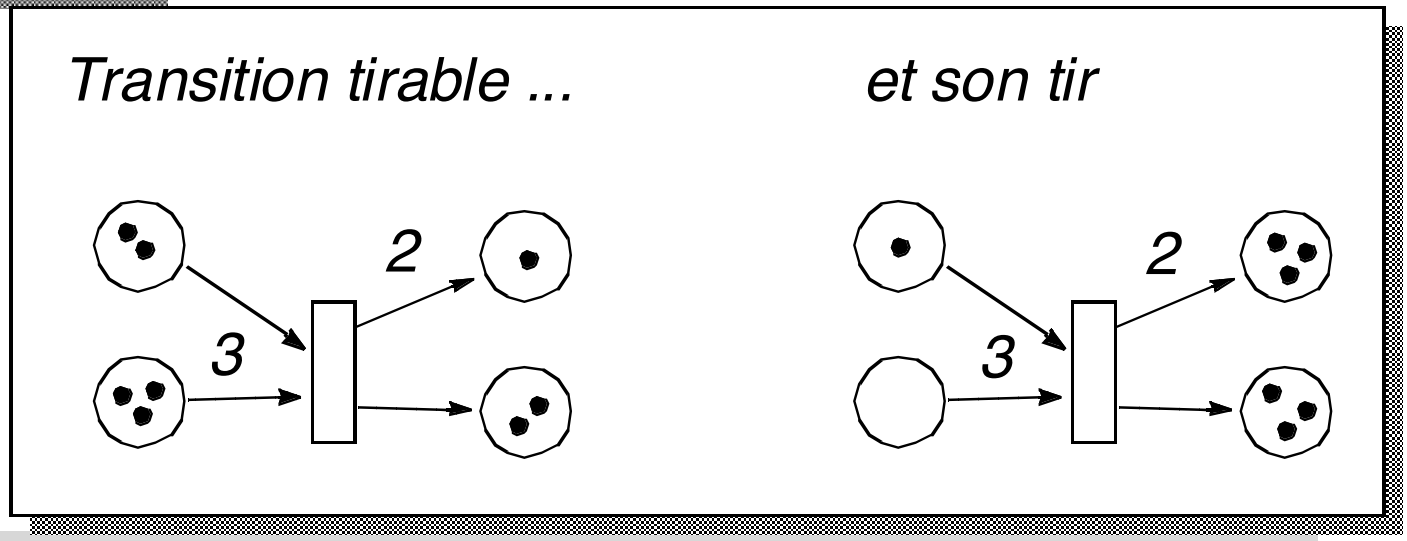
et son tir



(Cont'd)

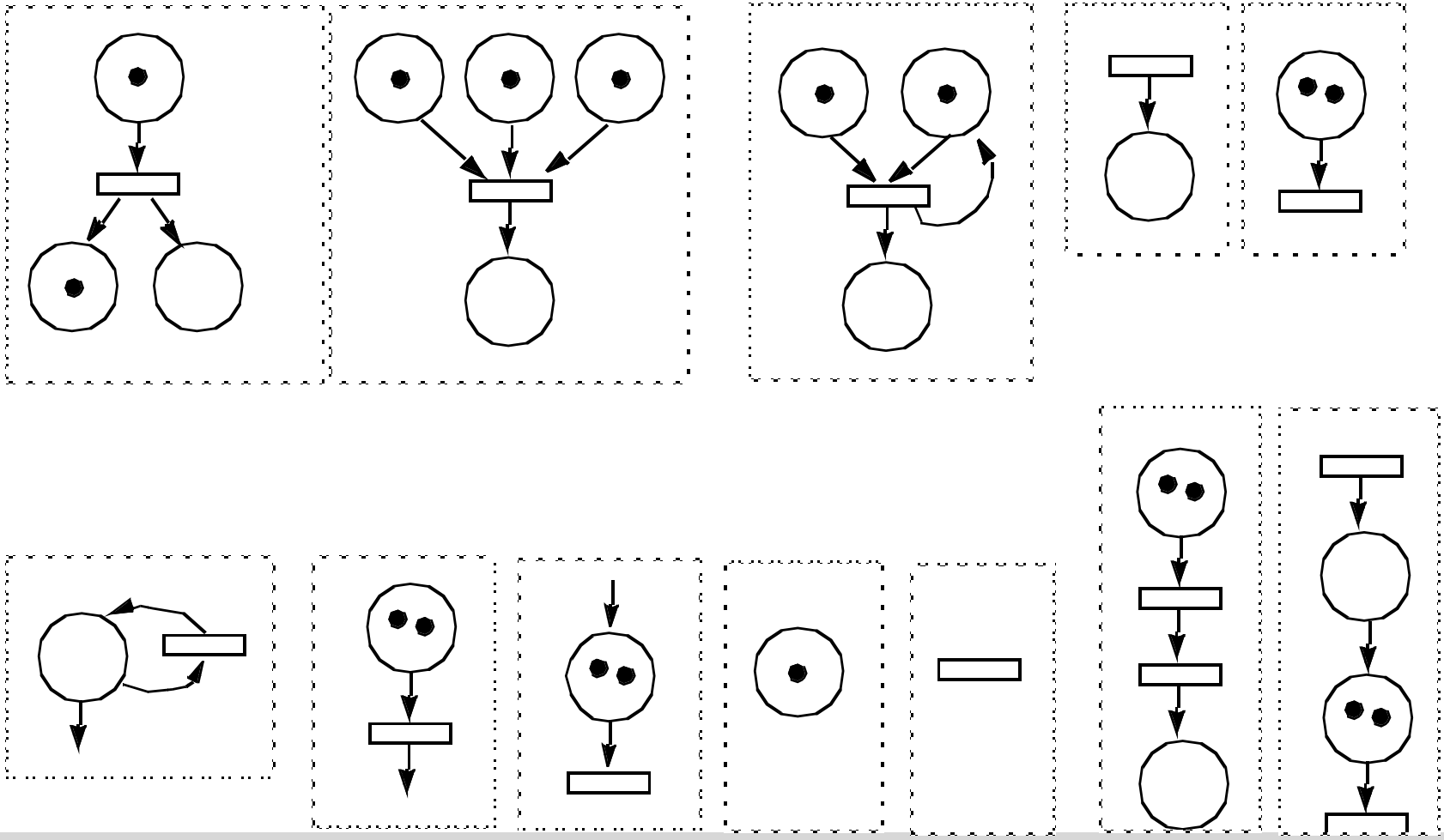
- Tir d'une transition avec valuation des arcs





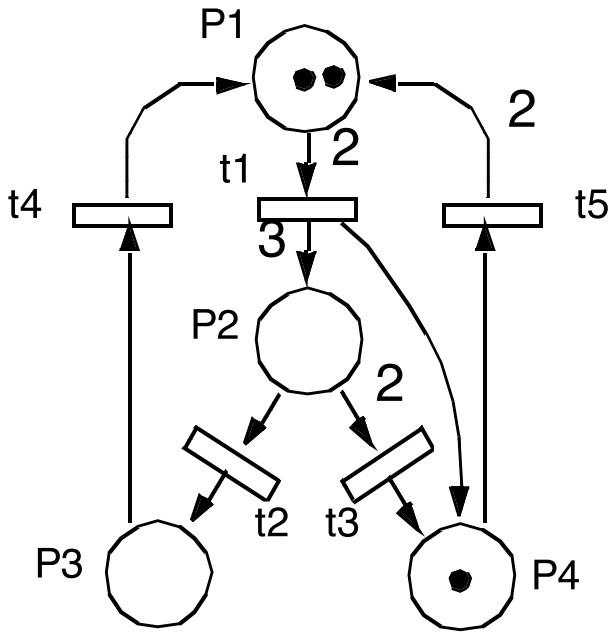
Exercice

Ces réseaux sont-ils des réseaux de Petri (rdP)? Pour chaque rdP indiquer les transitions tirables, le résultat après le tir et les transitions encore tirables après les franchissements.



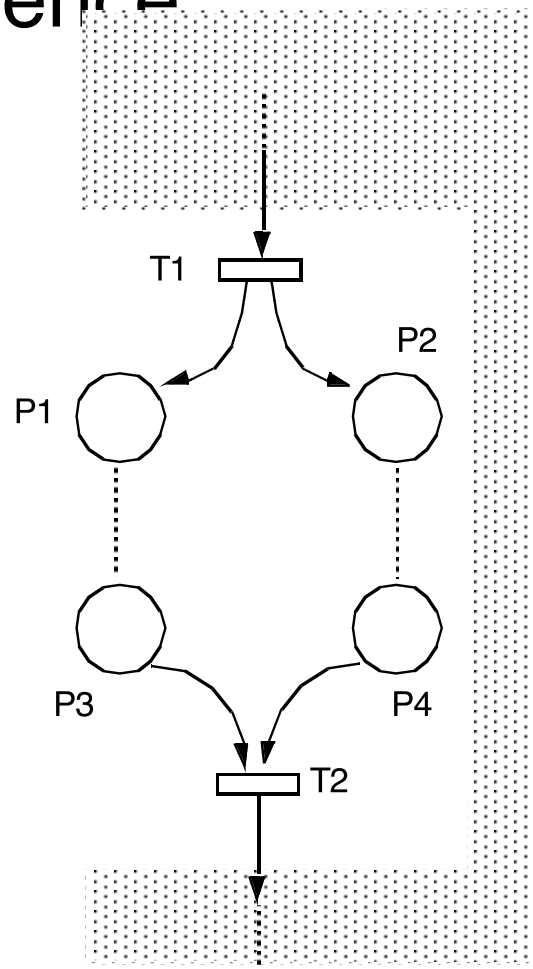
Exercice

Quelles sont les transitions tirables? Quelles seraient la distribution des jetons après le franchissement de chacune de ces transitions?



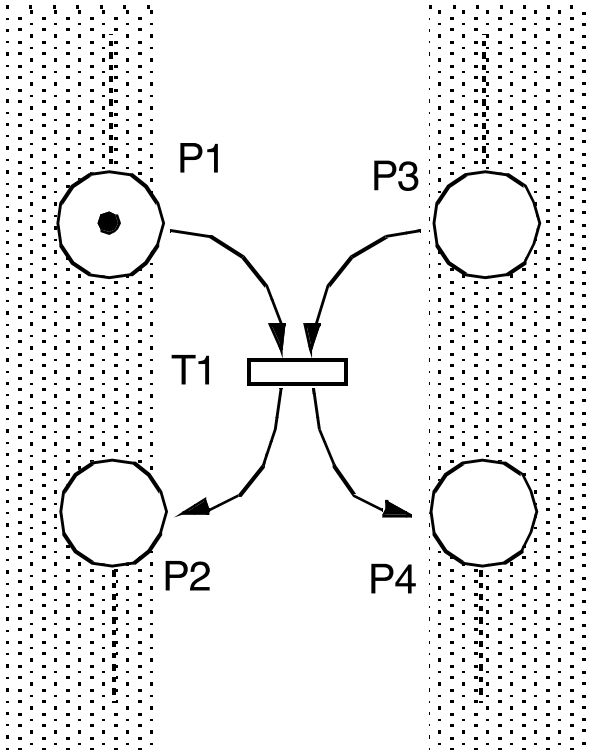
Schémas de synchronisations élémentaires

- Parallélisme ou concurrence

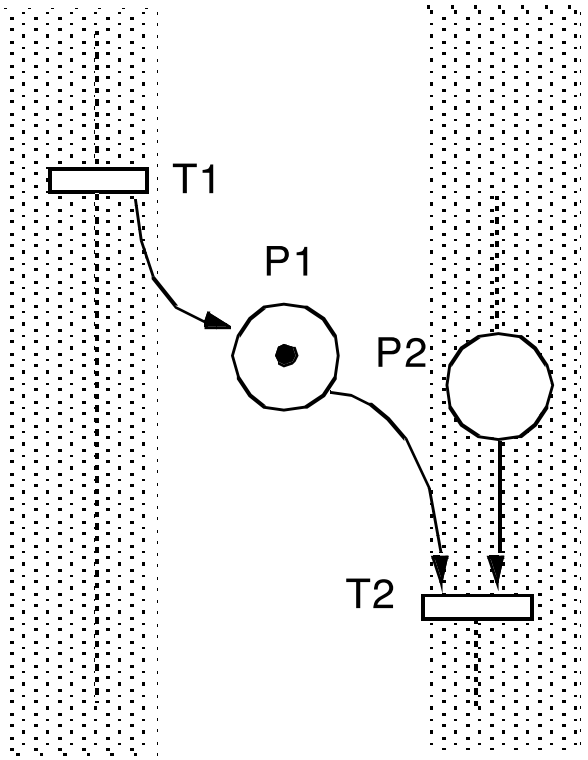


(Cont'd)

- Synchronisation



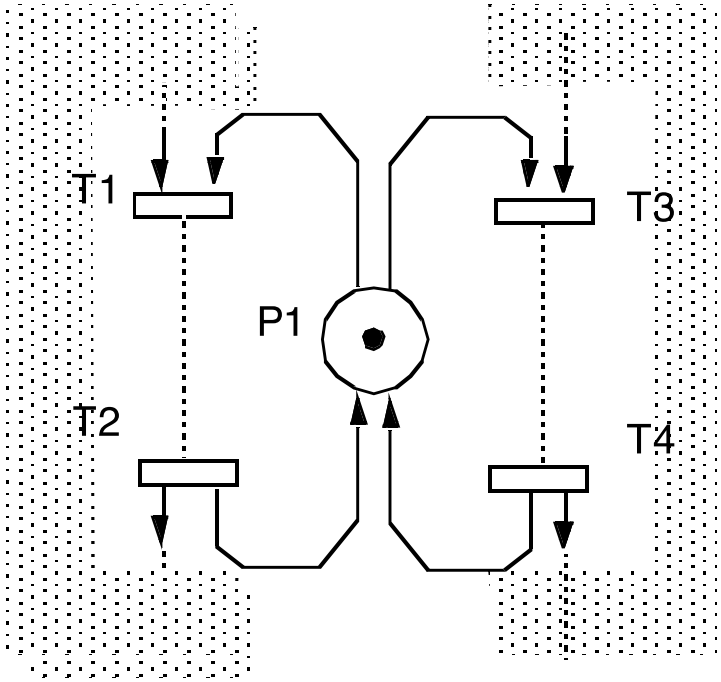
Rendez-vous



Sémaphore

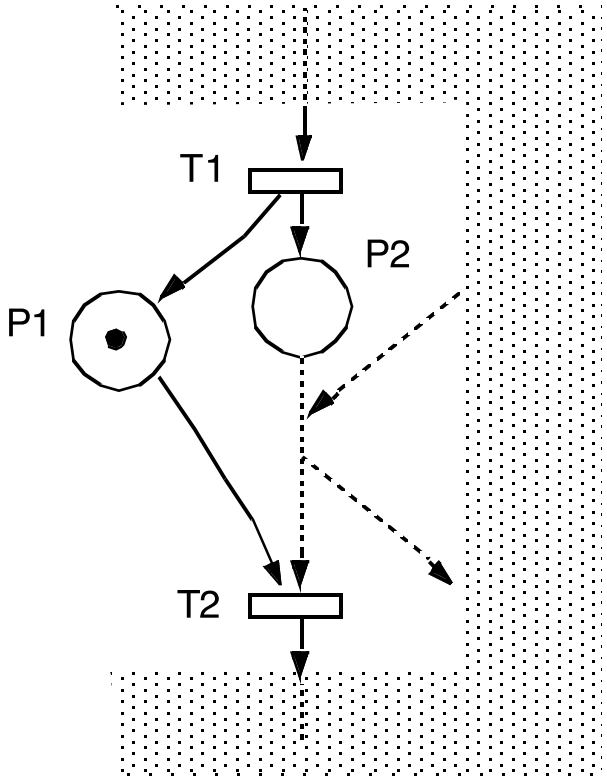
(Cont'd)

- Partage de ressource (exclusion, conflit, ...)

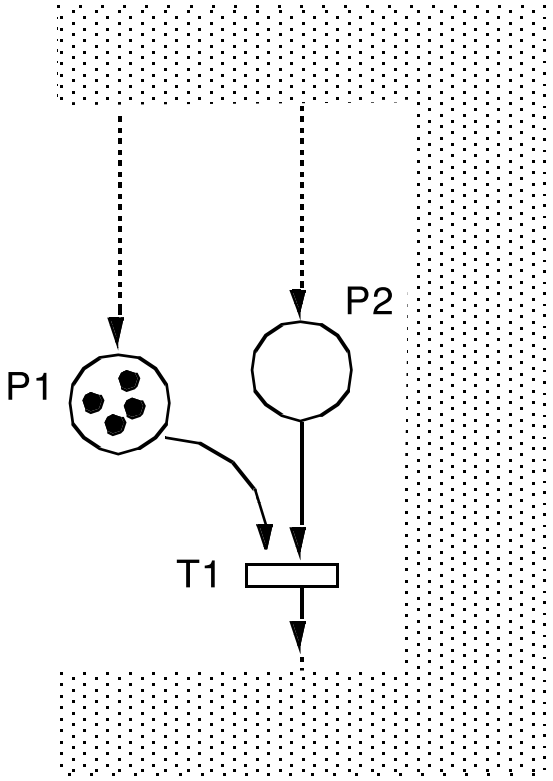


(Cont'd)

- Mémorisation



Franchissement



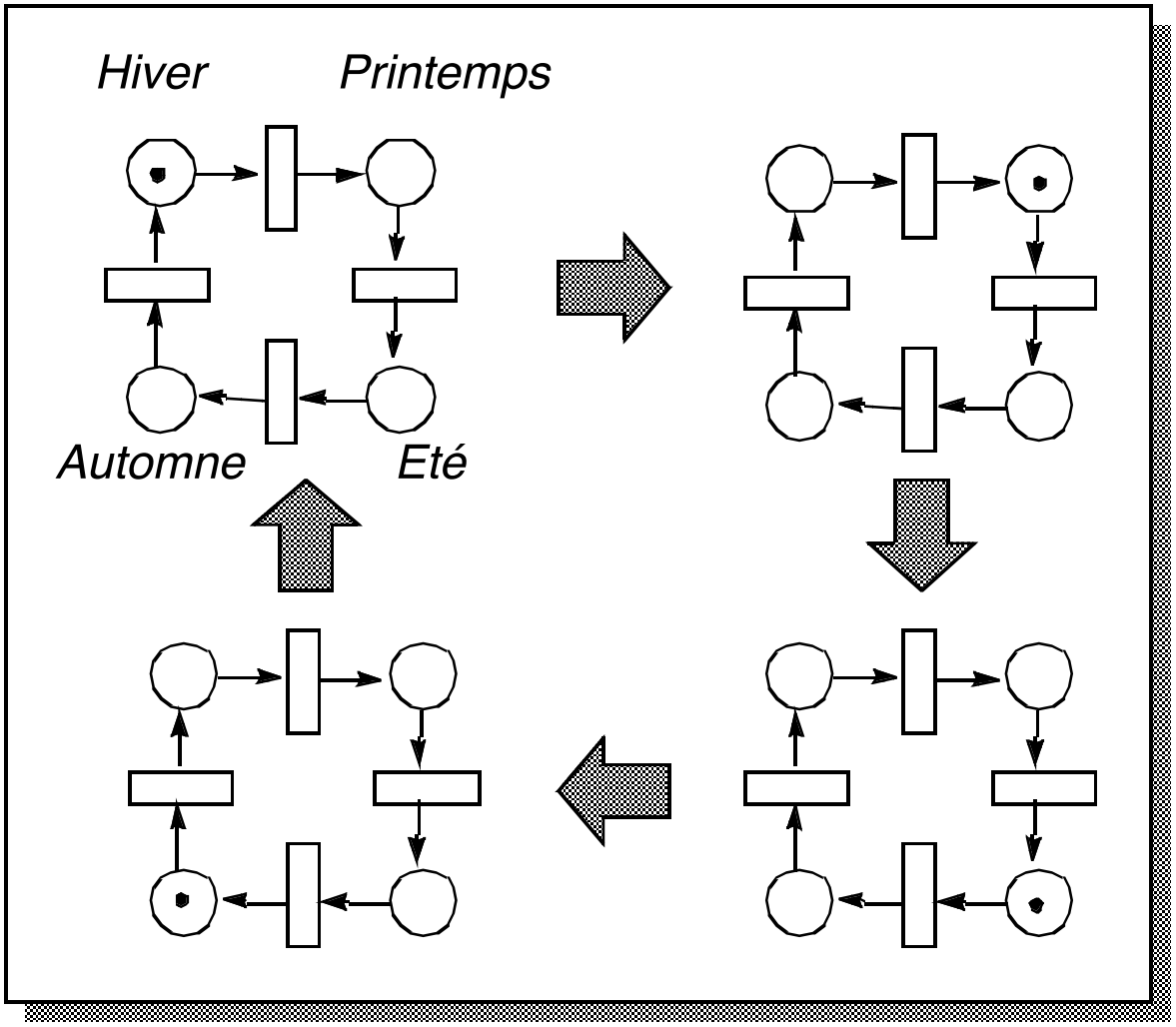
Nombre

Exercice

- Ordonnancement de tâches (PERT)
  - On définit six tâches dont les exécutions sont conditionnées par les règles suivantes.
  - Au départ, la tâche 1 est exécutable. Les tâches 2 et 3 ne peuvent être exécutées qu'après la fin de la tâche 1 (attention, cela ne signifie pas que l'exécution de ces tâches commence immédiatement à la fin de la tâche 1, ni même qu'elles commencent simultanément). La tâche 4 ne peut être exécutée qu'après la tâche 3, la tâche 5 après les tâches 2 et 4, et la tâche 6 après les tâches 4 et 5.
  - Enfin la tâche 1 ne peut être réexécutée qu'après la fin de la tâche 2, la tâche 3 après les tâches 1 et 6, et le cycle recommence indéfiniment.
  - *Remarque*: Si une tâche j ne peut être exécutée qu'après la fin de la tâche i, et si i est exéctuée plusieurs fois et successivement sans exécution de j, alors j peut ensuite être exécutée une ou plusieurs fois successivement.
- Représenter l'enchaînement de ces tâches par un rdP.

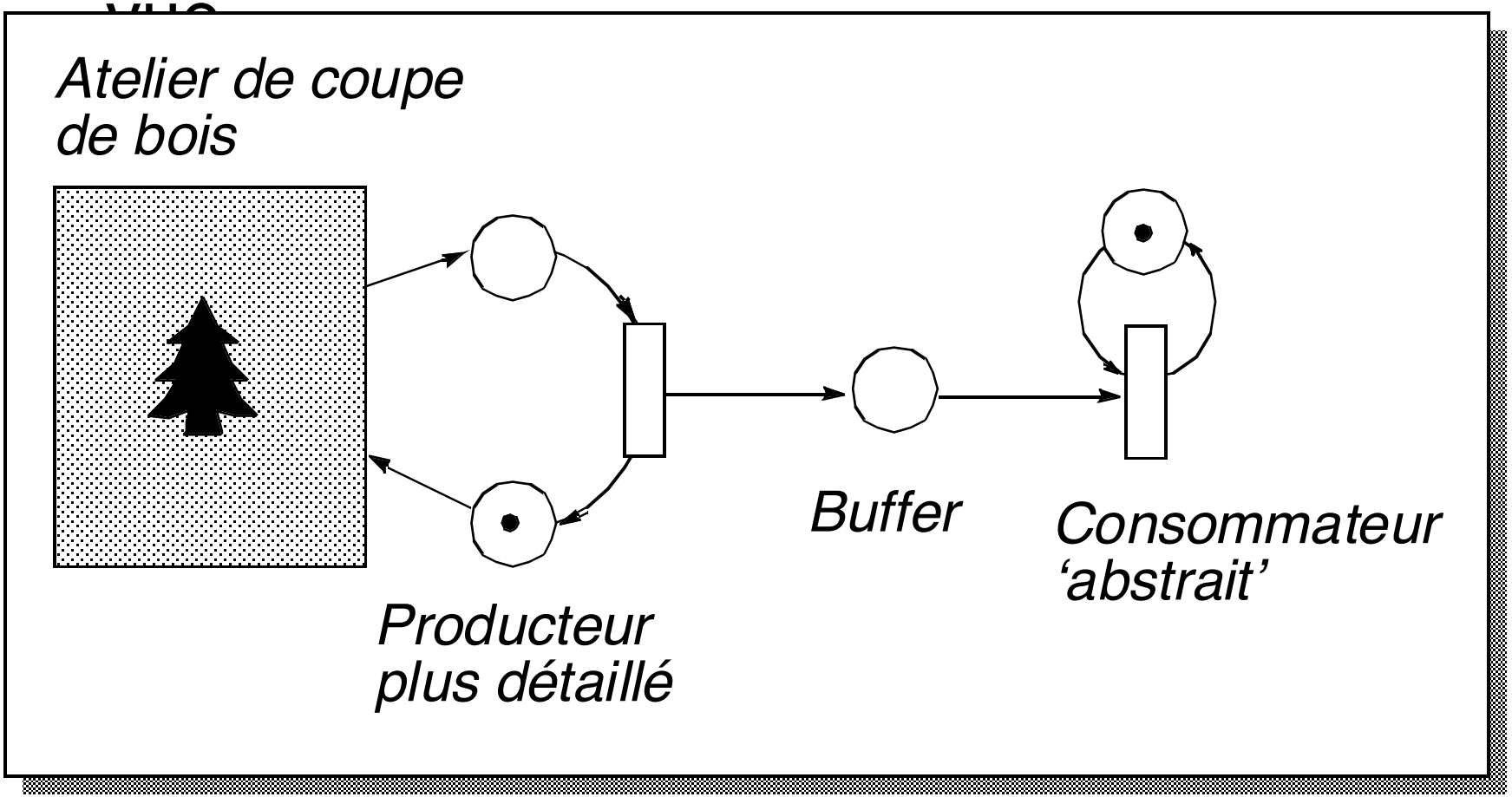
Reprise des propriétés principales d'un rdP

- Simulable



(Cont'd)

- Différents niveaux d'abstraction/points de vue

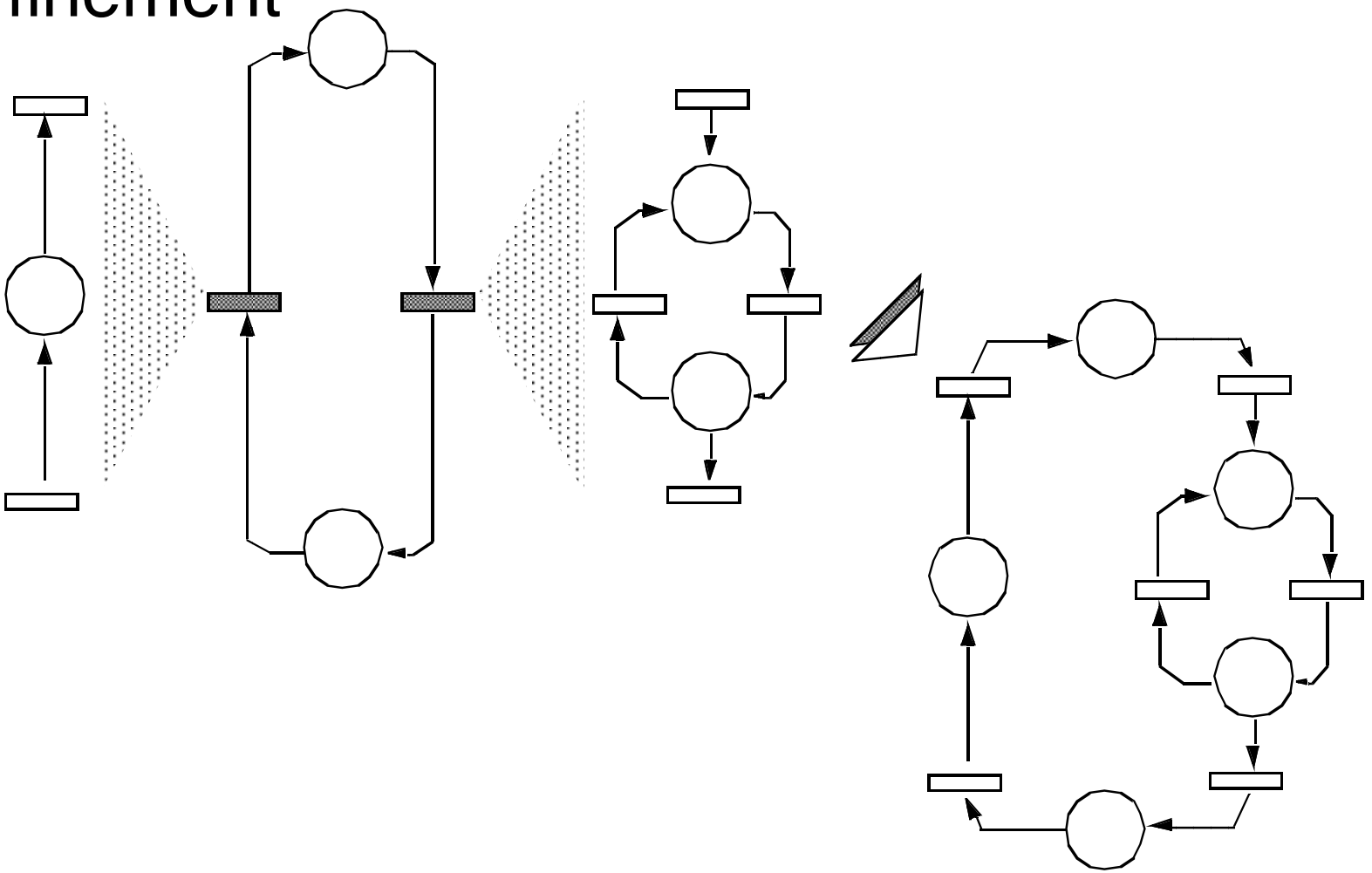


Structuration

- Approche par affinements successifs
- Approche par composition de réseaux
  - Sous-systèmes avec places partagées
  - Sous-systèmes avec une ou plusieurs transitions communes

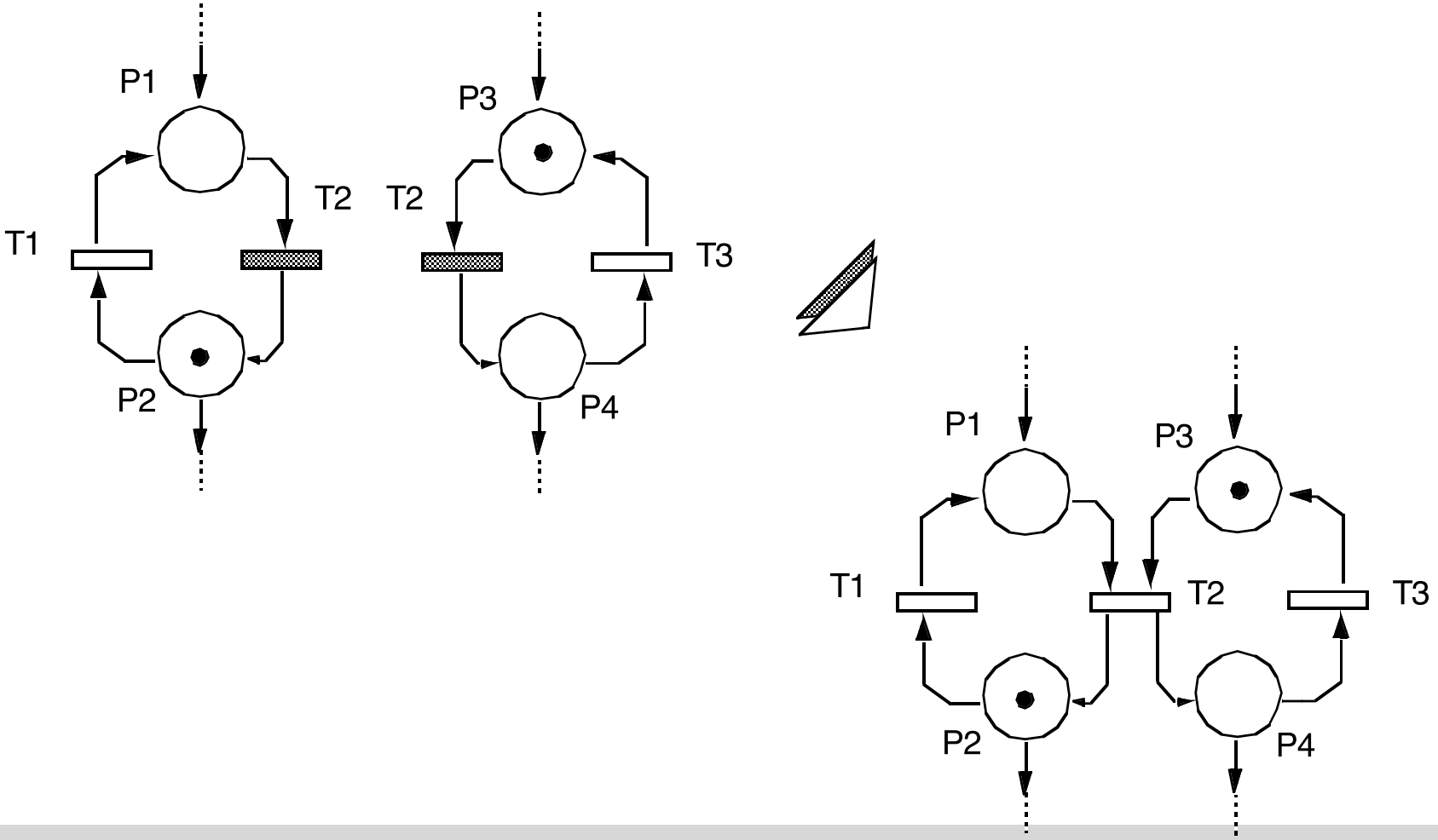
(Cont'd)

- Affinement



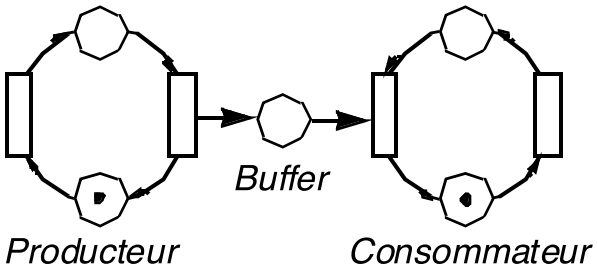
(Cont'd)

- Composition



Exercice

- Producteur-Consommateur avec un ‘buffer’ de capacité limitée, n
  - Le problème du producteur-consommateur met en jeu également une ressource partagée qui est un tampon (buffer). Le producteur crée des objets qu’il place dans le tampon et le consommateur attend l’existence d’objets dans le tampon. Quand il y a un objet, le consommateur le retire du tampon en le consommant.
  - Lorsque la capacité du tampon est illimitée, il y a une seule synchronisation: celle du consommateur qui pour consommer doit attendre que le tampon ne soit pas vide.
  - Avec une capacité limitée à *n* objets, il y a une synchronisation supplémentaire: celle du producteur qui doit attendre si le tampon est plein.



Exercice

Lecteur-Rédacteur

Dans ce problème, il y a deux types de processus: les processus lecteurs et les processus rédacteurs. Tous ces processus partagent une ressource commune comme, par exemple, un fichier. Les processus lecteurs ne modifient pas cette ressource alors que les rédacteurs la modifient. Donc les processus rédacteurs doivent être en exclusion mutuelle avec les autres processus rédacteurs et les processus lecteurs. Par contre les processus lecteurs peuvent accéder simultanément à la ressource.

- Modéliser ce problème en supposant que le nombre de lecteurs est borné par *n*. Il y a *s* lecteurs et *t* rédacteurs.
- Peut-on résoudre le problème si l'on ne connaît pas une borne supérieure au nombre de lecteurs?

Résumé

- Les réseaux de Petri servent à la modélisation et à l'analyse de systèmes dynamiques discrets.
- Système conditions-événements: représentation par places, transitions et jetons.
- Système ressources: plusieurs jetons par place, valuation des arcs.