

XARXES DE PETRI

Formalització de models de simulació

Carl Adam Petri

- Carl Adam Petri (12 de juliol de 1926, 2 de juliol de 2010). Matemàtic i informàtic nascut a Leipzig.
- Va desenvolupar les xarxes de Petri el 1962, com a part de la seva dissertació, “Kommunikation mit Automaten”. Va treballar del 1959 al 1962 a la universitat de Bonn i va rebre el seu PhD al 1962 de la universitat de tecnologia de Darmstadt.



Standardization

- <http://www.petrinets.info/>
- <http://www.pnml.org/>

2011-02-14	Standard published: ISO/IEC 15909-2:2011 Systems and software engineering - High-level Petri nets - Part 2: Transfer format. Available from ISO [Editor's Announcement]
2005-06-23	New Working Draft of ISO/IEC 15909-2 Systems and software engineering - High-level Petri nets - Part 2: Transfer format submitted for a combined ISO/IEC SC7 WD/CD registration and CD ballot. Comments welcomed - formal or otherwise. [Editor's Announcement ISO/IEC 15909-2 WD (Version 0.9.0)]
2004-12-02	Standard published: ISO/IEC 15909-1:2004 Systems and software engineering - High-level Petri nets - Part 1: Concepts, definitions and graphical notation. Available from ISO .
2003-02-27	International Standard Ballot of ISO/IEC 15909-1. [Ballot ZIP.]
2002-11-22	Combined WD/CD Registration/FCD ballot for ISO/IEC 15909-1. [FCD PDF, Disposition RTF.]
2002-10-21	Great news : Green Light for High-level Petri Net Standard. [Letter Ballot Summary PDF.]
2002-09-11	Notification of ballot for reinstatement of Standard.
2002-05-12	Standard unintentionally cancelled during Draft International Standard (DIS) phase!
2000-10-28	Final Committee Draft Version 4.7.1 published. [Editor's Announcement FCD]
1998-01-28	Letter Ballot Summary of the Standard.
1997-10-02	Draft standard Version 3.4 published.
1997-08-14	Reply to Comments on Draft Petri net Standard V2.1.
1997-01-17	JTC1 News : Standard will not be Cancelled.
1996-12-13	Resolution 8.4 , 10th Plenary Meeting of ISO/IEC JTC 1.
1996-11-29	Australian Response to Cancellation Proposal.
1995-08-14	Petri net Subdivision Proposal .

Xarxes de Petri

- Una xarxa de Petri és un graf dirigit, ponderat i bipartit.
- Té dos tipus de nodes:
 - ▣ Llocs: Representats per cercles.
 - ▣ Transicions: Representats per rectangles.
- Els ARCS poden unir nodes llocs a nodes transicions o al revés.

Xarxes de Petri

- $RdP = (P, T, A, W, M_0)$
- $P = \{T_1, T_2, T_3, \dots, T_n\}$: conjunt de nodes de tipus lloc
- $A \subset (P \times T) \cup (T \times P)$: conjunt d'arcs de la xarxa = subconjunt del producte cartesià de tots els nodes P i T .
- $W = a \rightarrow \{1, 2, 3, \dots\}$: Pes associat a cada arc.
- $M_0 = P_i \rightarrow \{1, 2, 3, \dots\} \forall P_i$: nombre de marques inicials en cada node tipus lloc (marcatge inicial)

Xarxes de Petri

- $E(T_i) = \{P_i \in P, (P_i, T_i) \in A\}$: conjunt de llocs d'entrada a la transició T_i .
- $S(T_i) = \{P_i \in P, (T_i, P_i) \in A\}$: conjunt de llocs de sortida per la transició T_i .

Xarxes de Petri

- Llocs representen:
 - ▣ Cues
 - ▣ Estats del sistema
 - ▣ Recursos del sistema.
- Transicions:
 - ▣ Esdeveniments del sistema.
- Arcs:
 - ▣ Solen tenir un pes associat. Permet descriure les condicions necessàries per tal que el esdeveniment en la transició es pugui activar.

Xarxes de Petri



- Esdeveniments associats a transicions.
- Activitats associades a llocs.
- Permeten descompondre el model.

Interpretació típica de transicions i llocs

Input Places	Transition	Output Places
Preconditions	Event	Postconditions
Input data	Computation step	Output data
Input signals	Signal processor	Output signals
Resources needed	Task or job	Resources needed
Conditions	Clause in logic	Conclusions
Buffers	Processor	Buffers

Xarxes de Petri

- Permeten realitzar un anàlisi del model, ajuden també en la:
 - ▣ Verificació.
 - ▣ Validació.
 - ▣ Anàlisi dels resultats de simulació.
- Pot representar formalment el paral·lisme i la sincronització.

Xarxes de Petri

- Es possible efectuar un anàlisi quantitatiu del sistema.
 - ▣ Estudiant aspectes estructurals
 - Situacions de bloqueig.
 - Abastabilitat de certs estats.
- Permet determinar tots aquells esdeveniments que poden aparèixer quan:
 - ▣ El sistema es troba en un determinat estat.
 - ▣ Es desencadena un determinat esdeveniment.

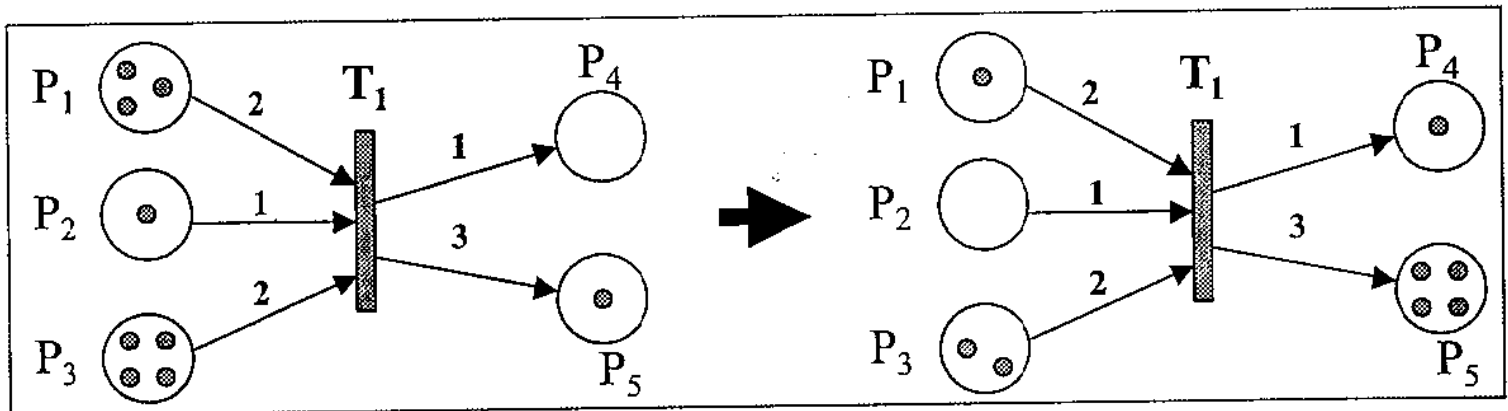
Xarxes de Petri

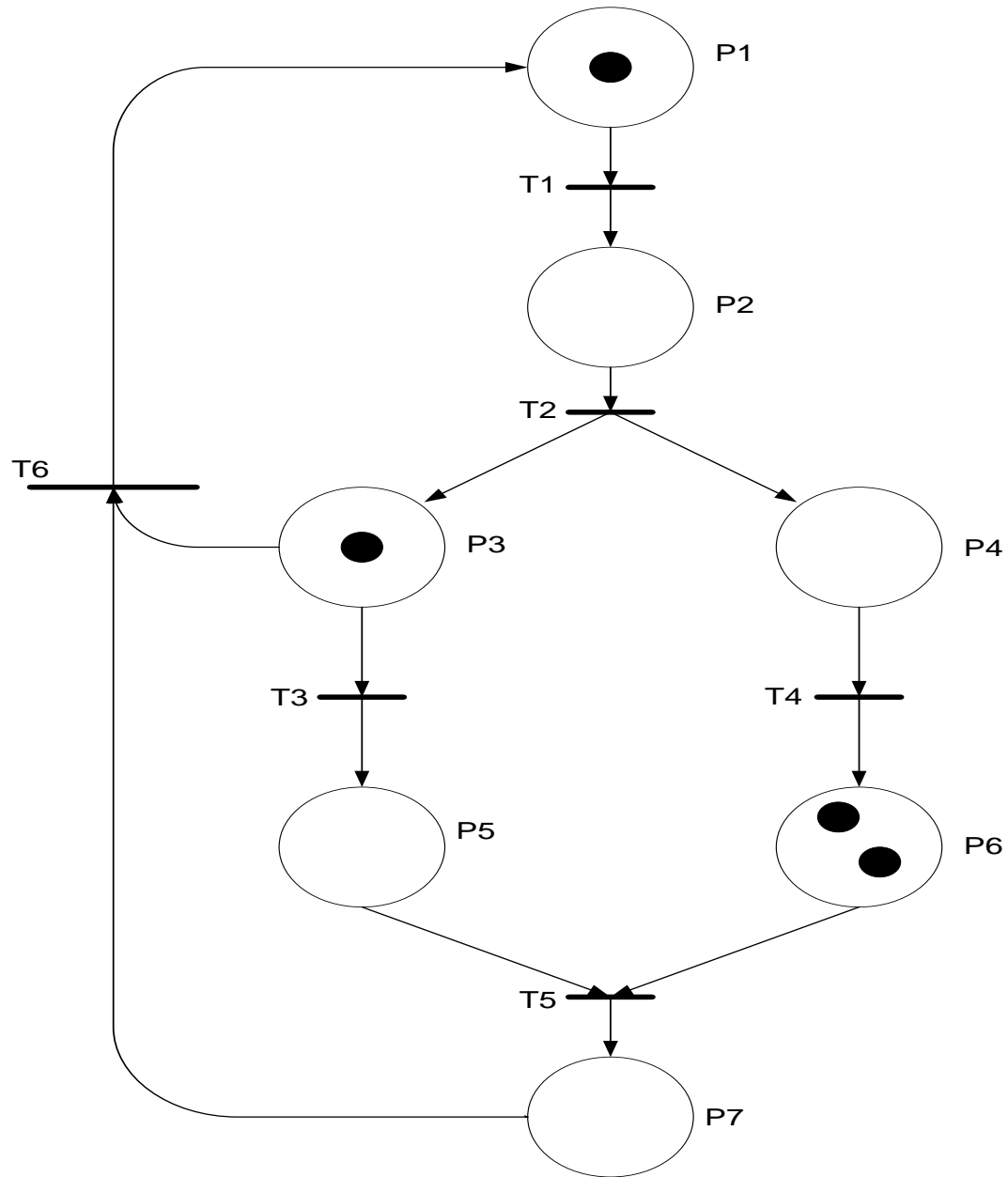
- Permet formalitzar en diferents nivells d'abstracció.
- Permet la descripció de un sistema complex a partir de la metodologia bottom-up.
- Es un formalisme de modelat gràfic amb molt poques regles sintàctiques.

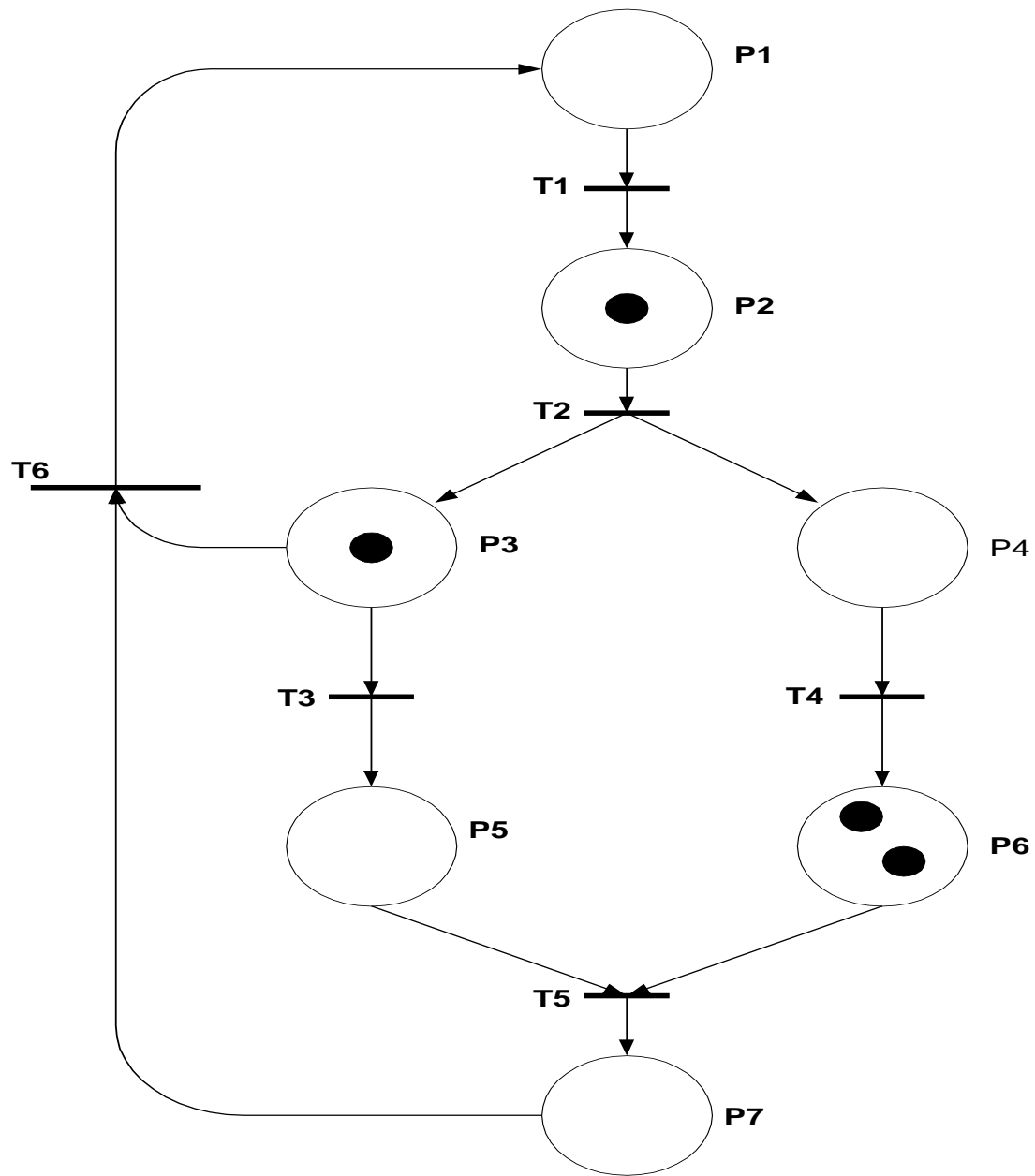
Programació Bottom-up

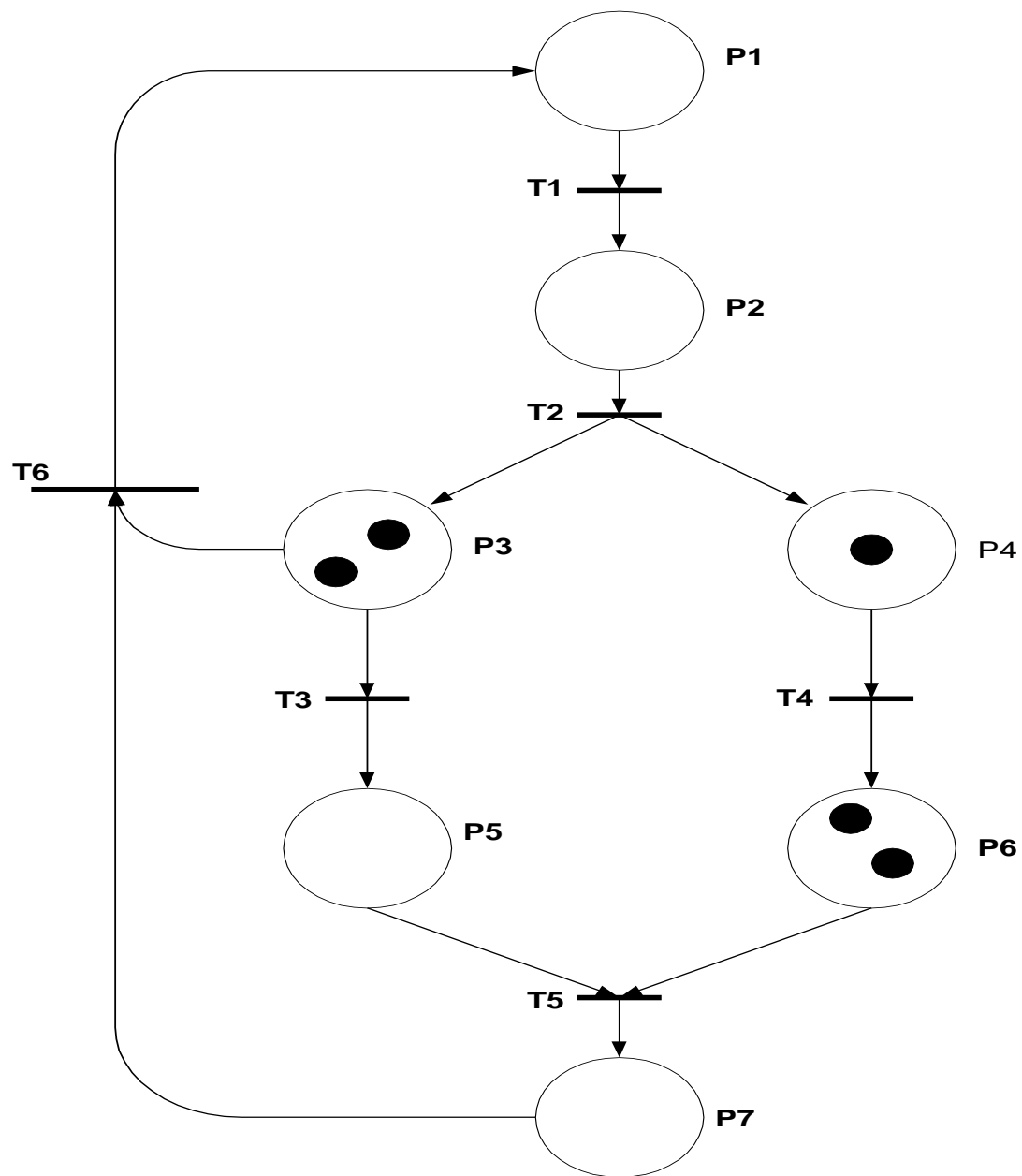
- Construcció del model del sistema complet a partir de les Xarxes de Petri (submòduls) dels sistemes que han estat prèviament desenvolupats i verificats.

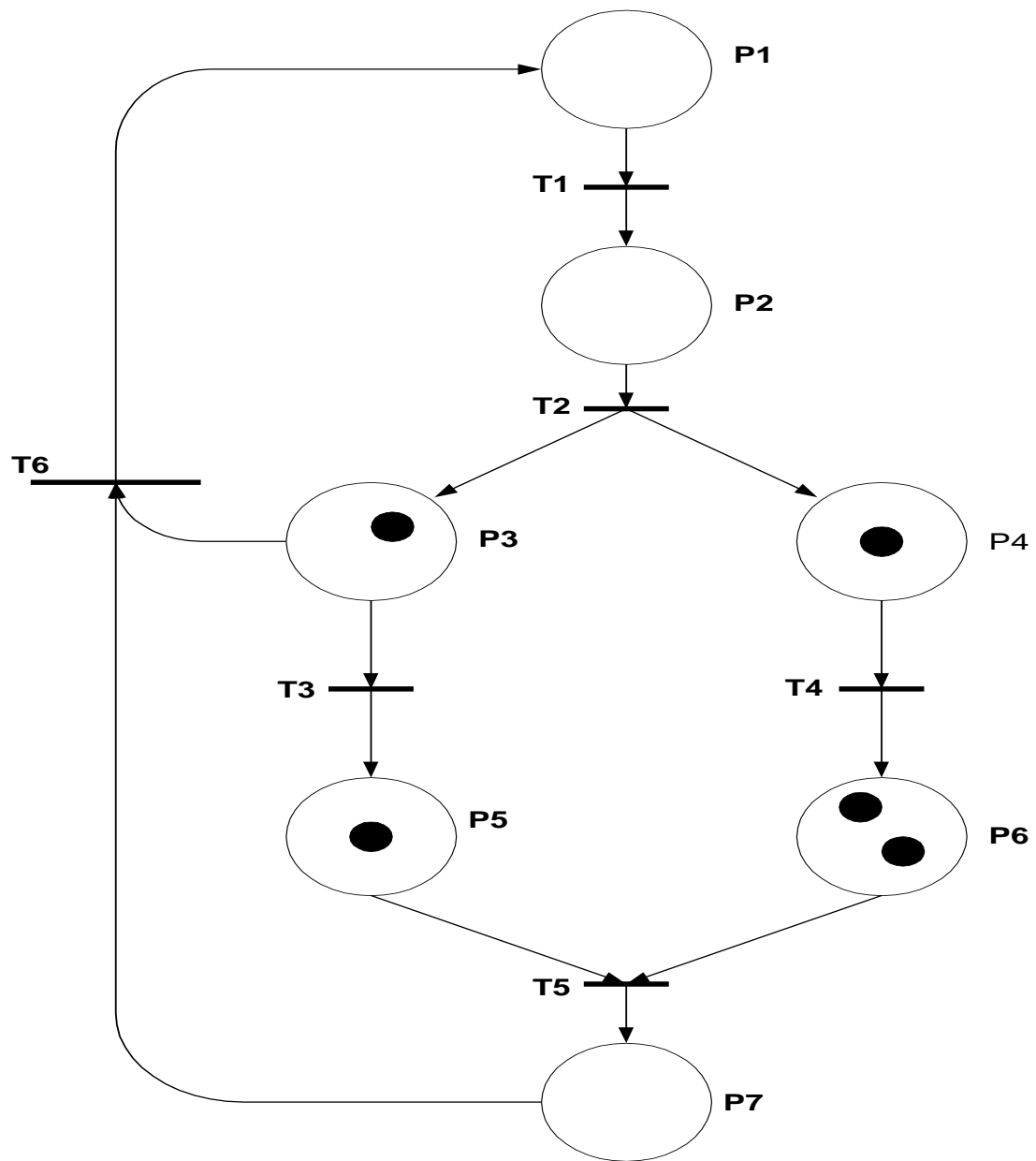
Xarxes de Petri

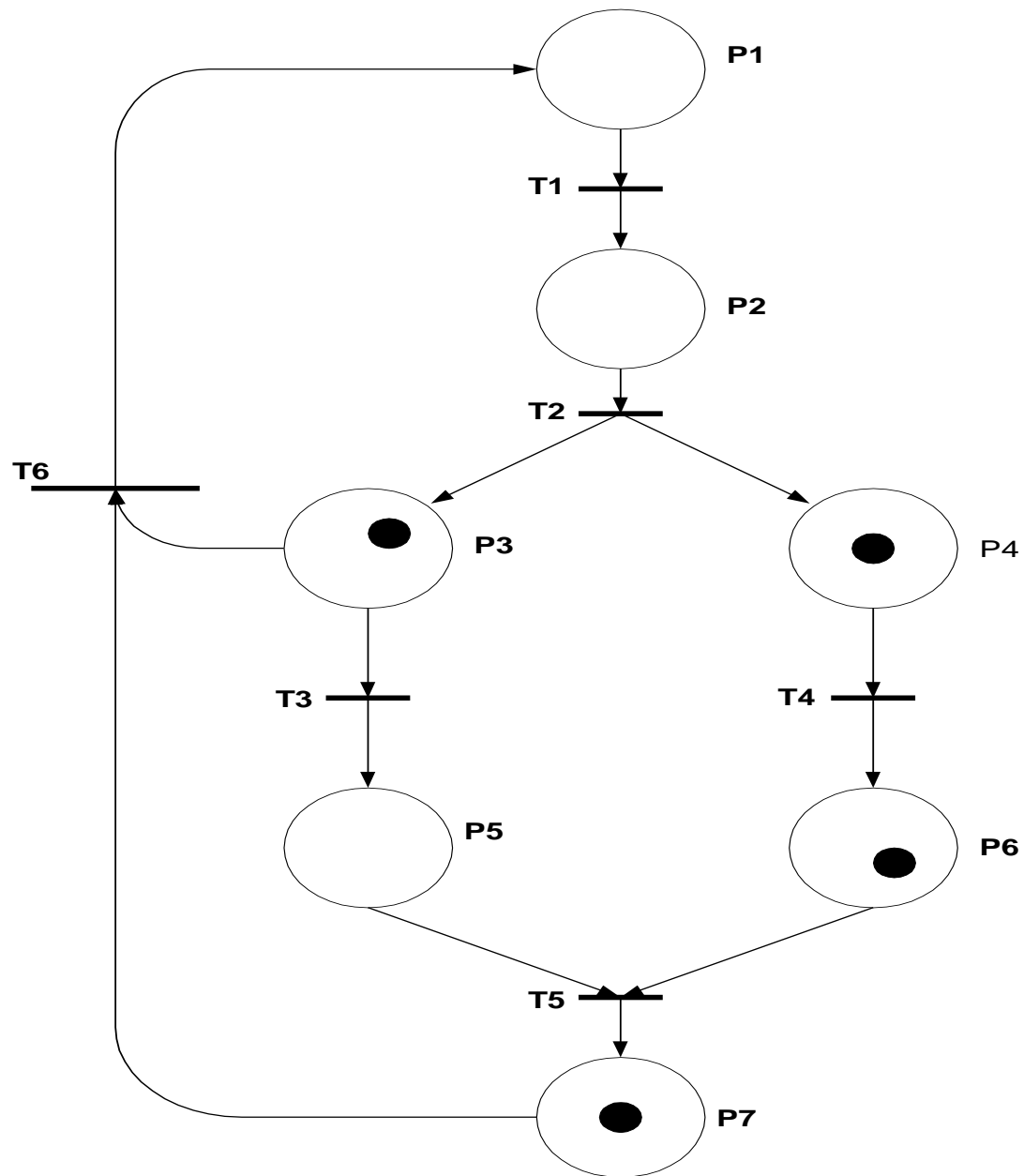


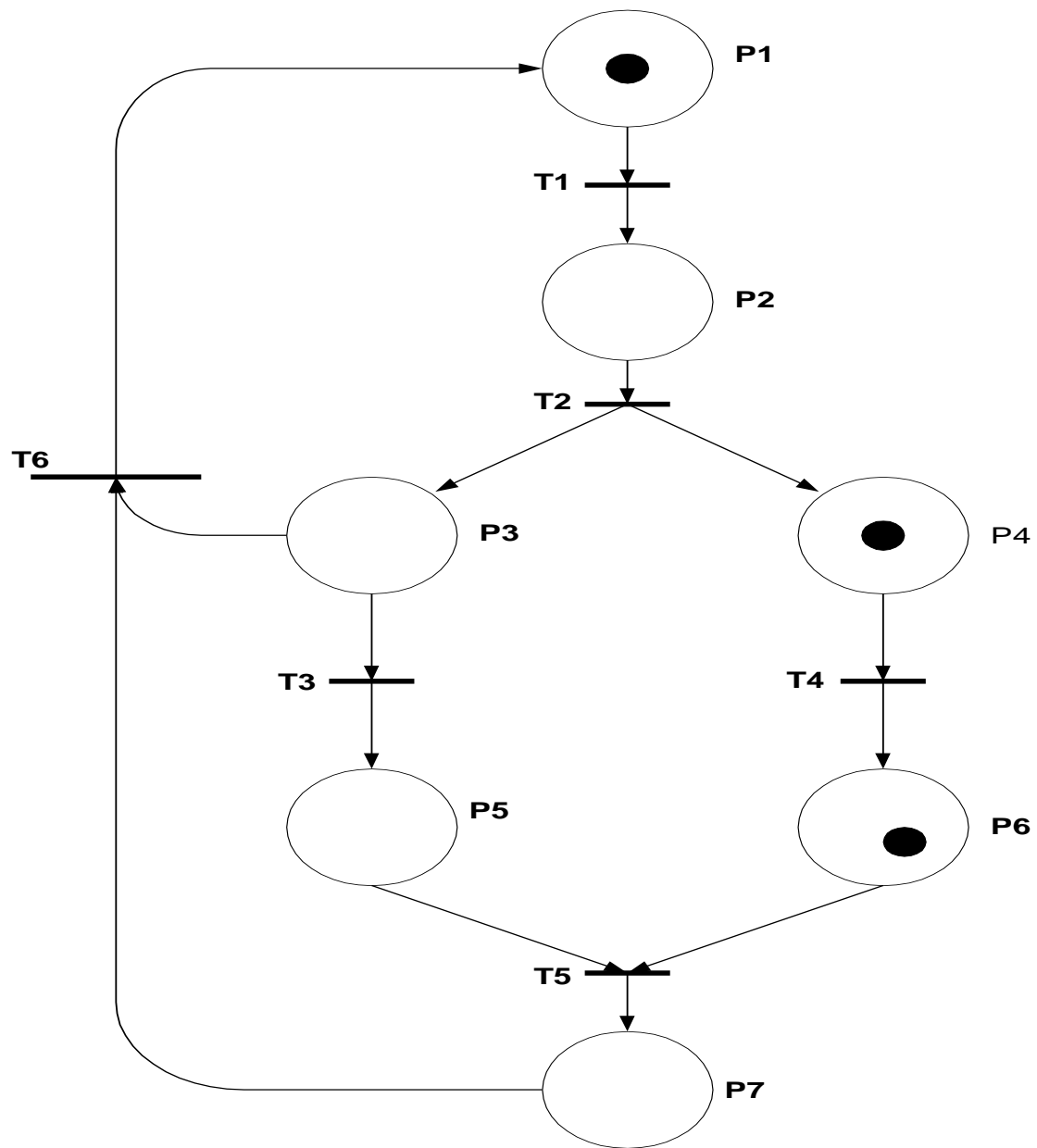


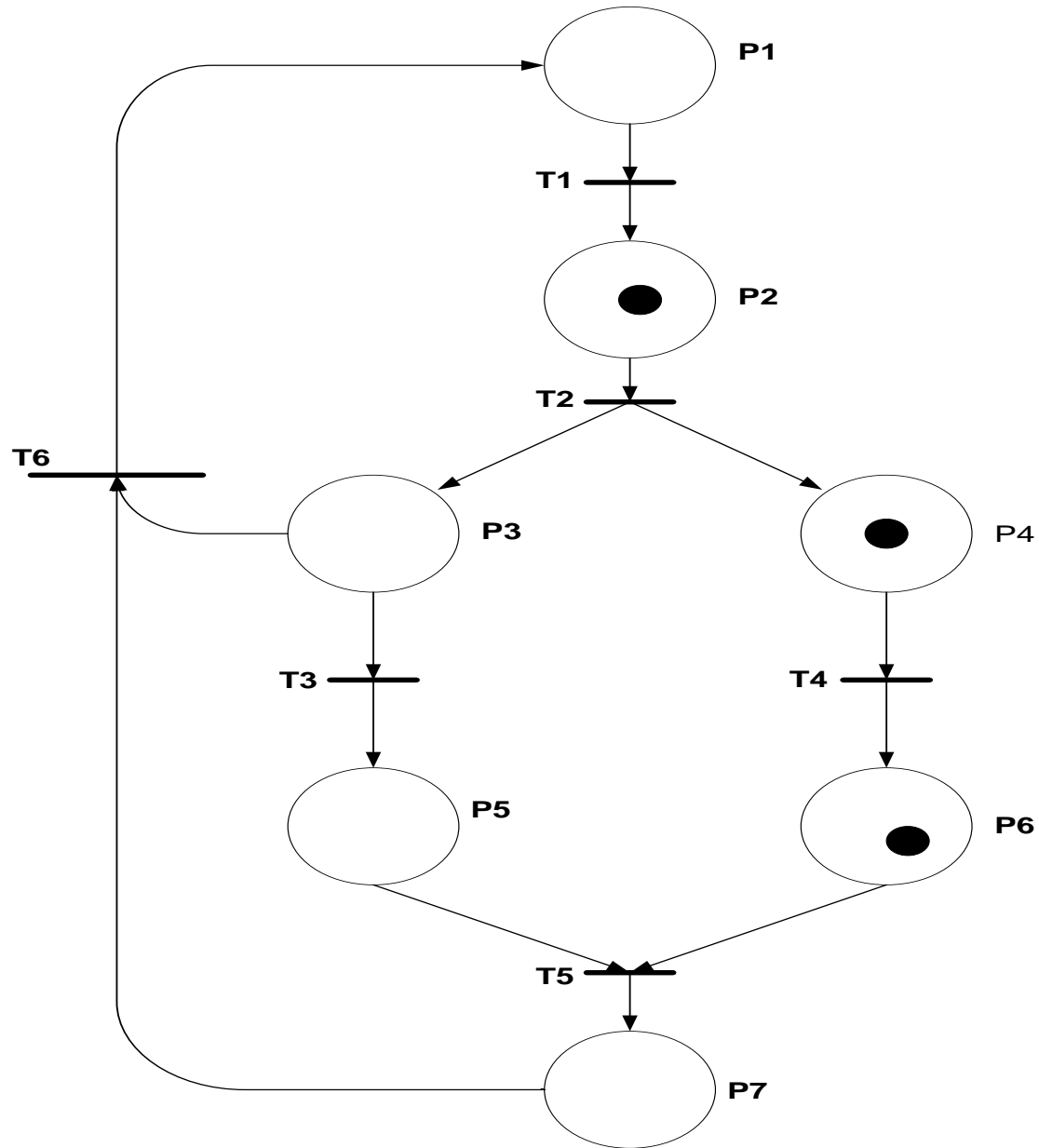


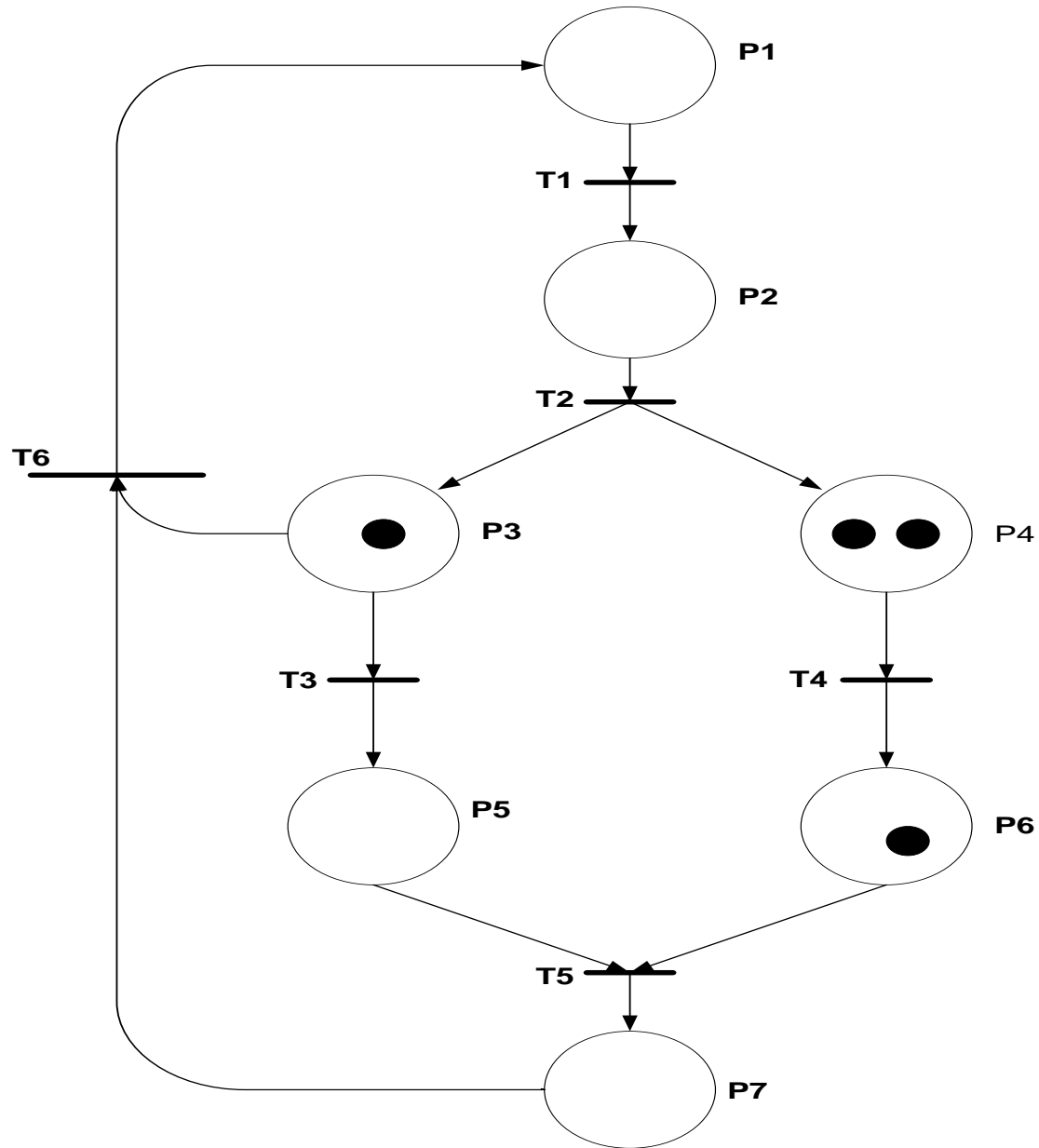


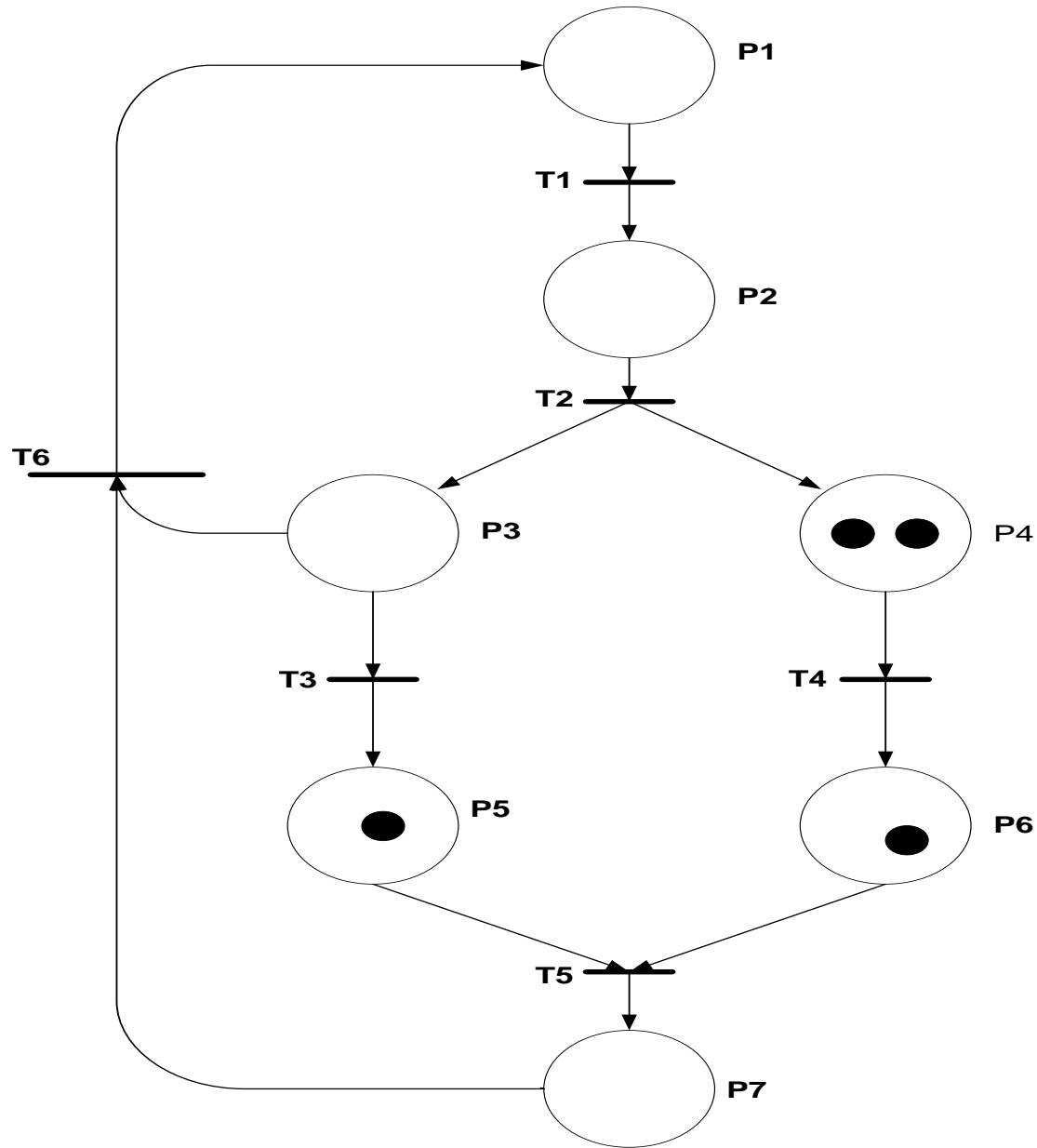


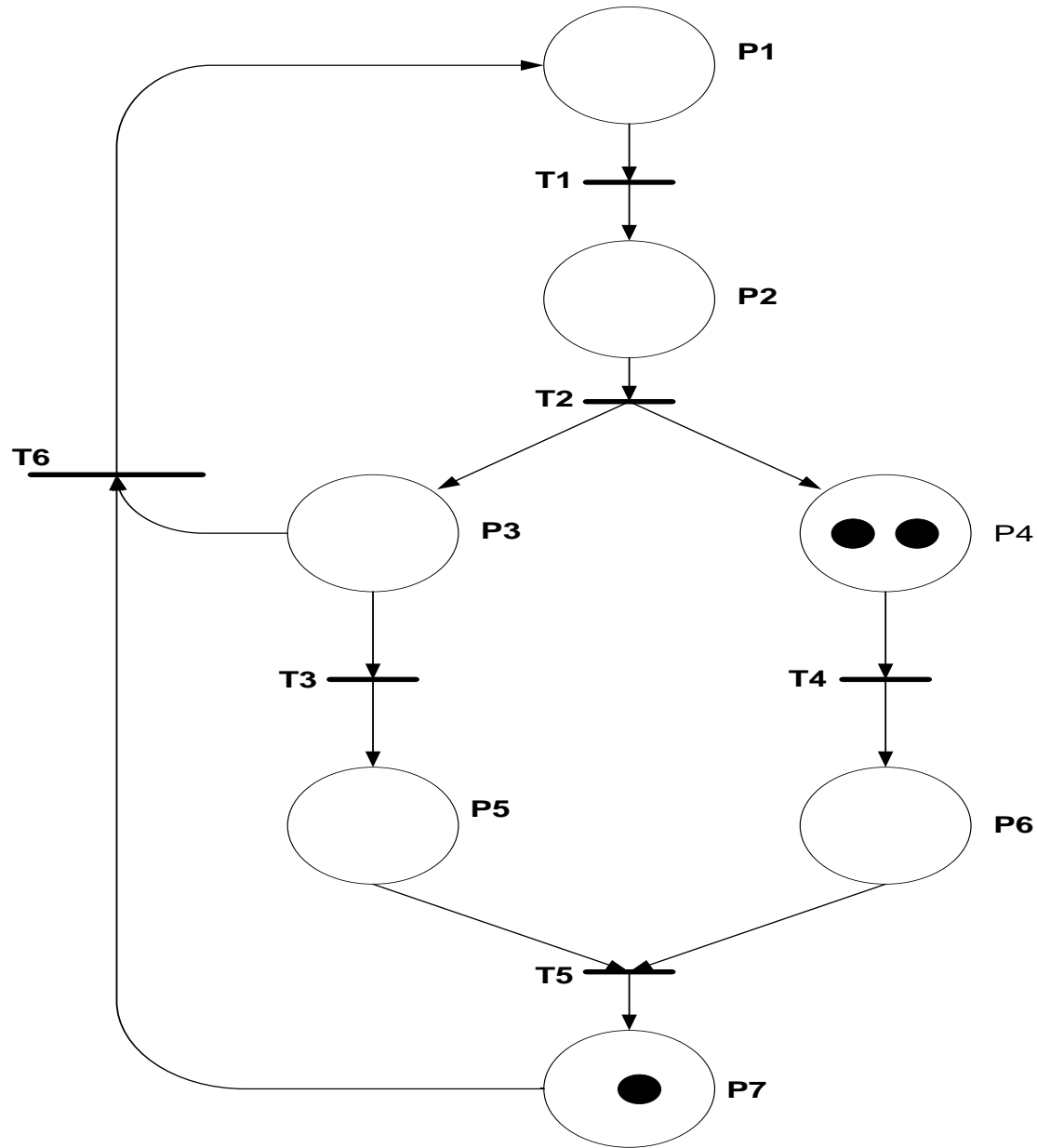


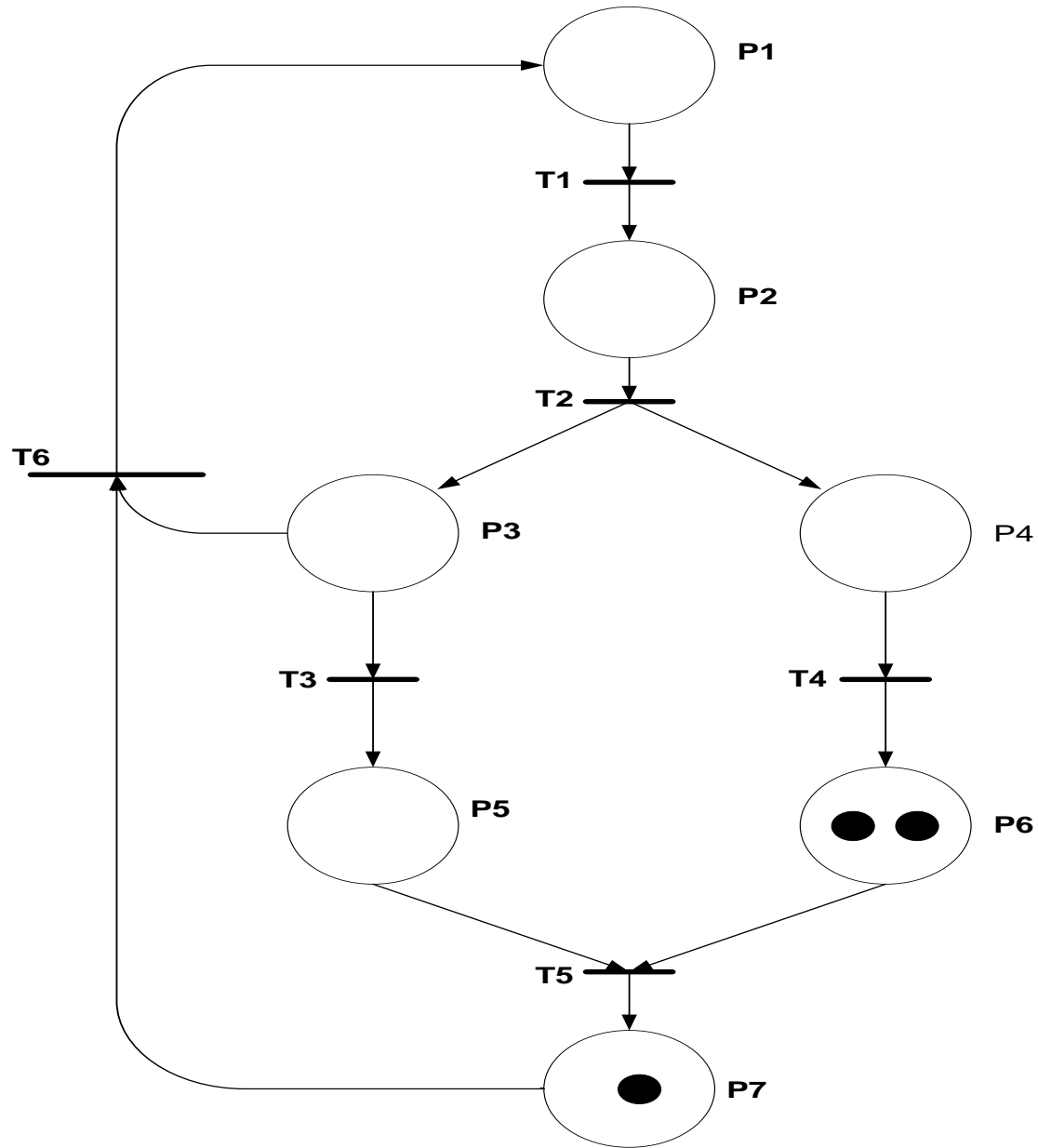












Xarxes de Petri

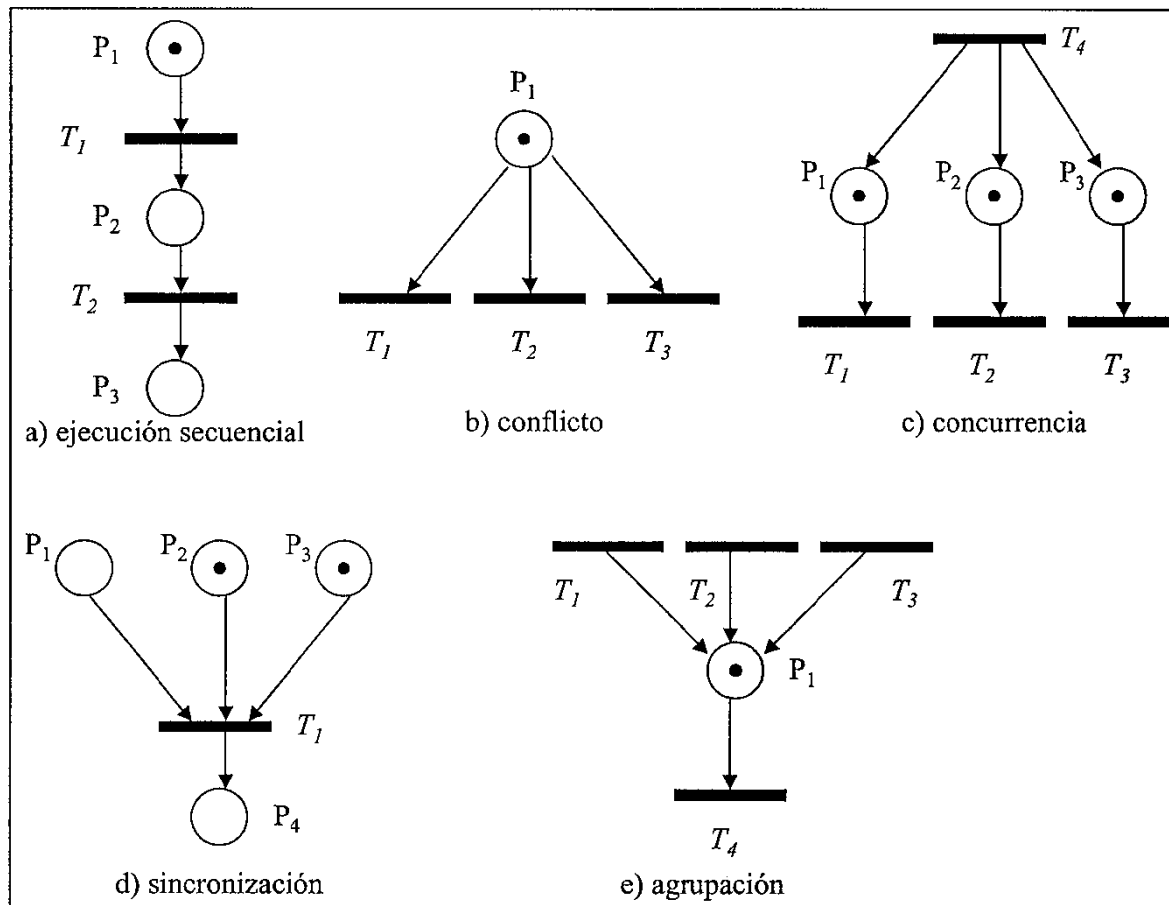
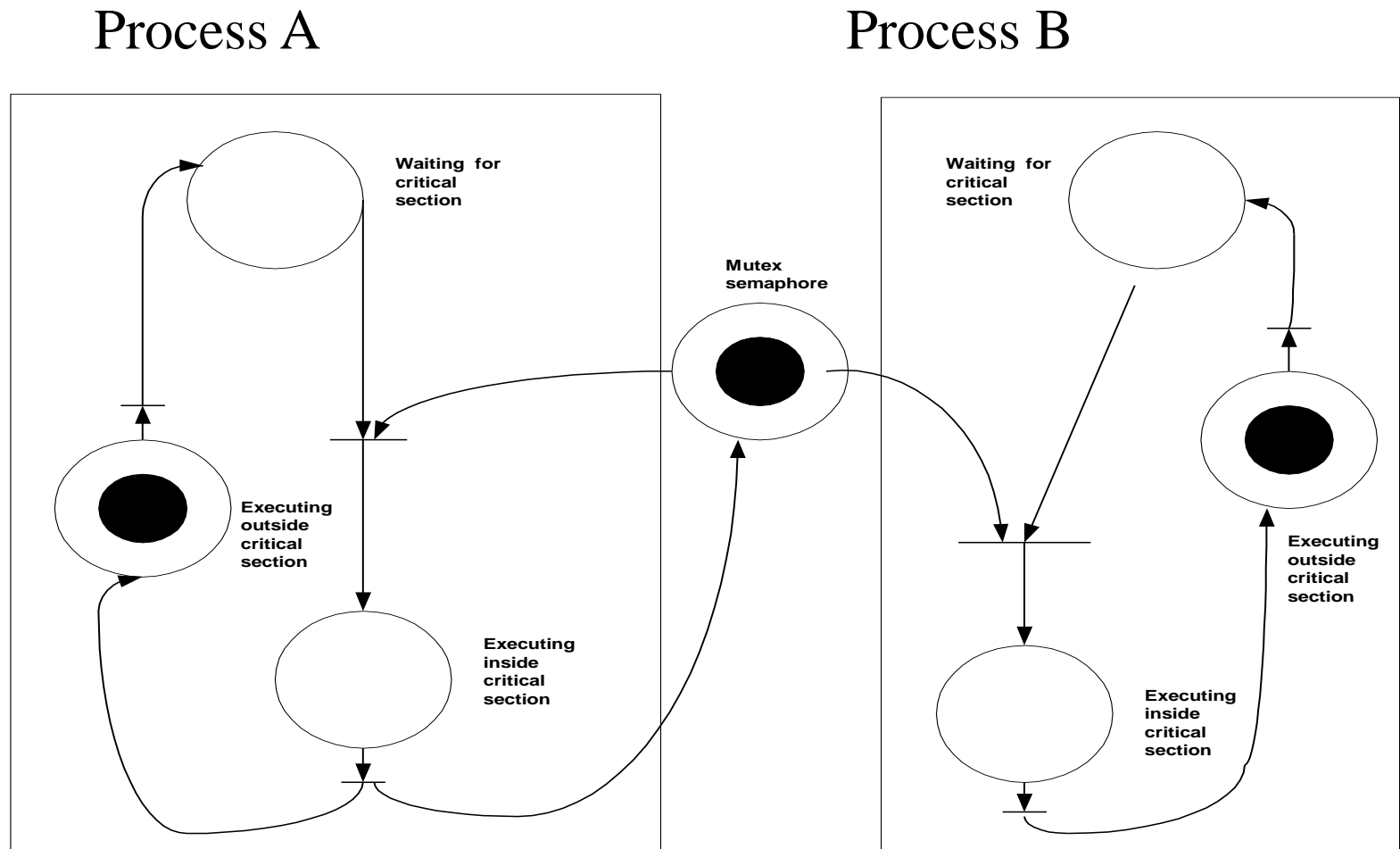


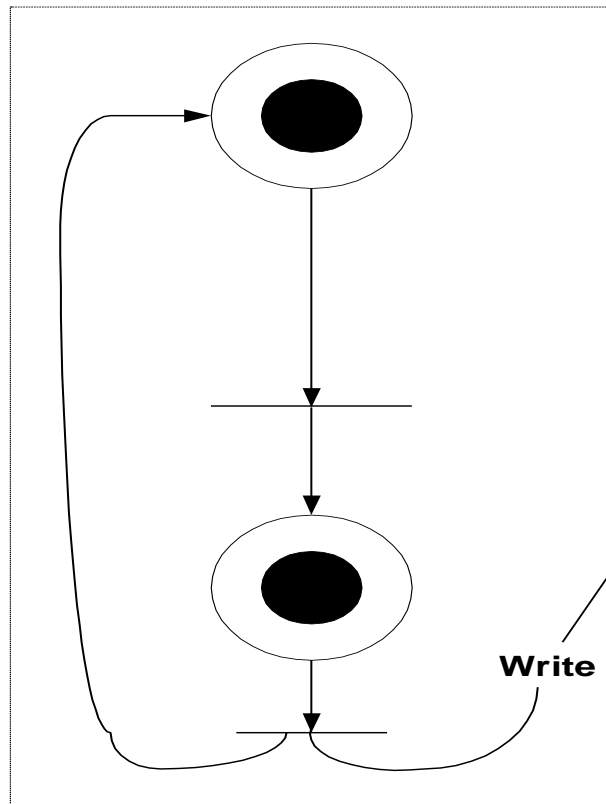
Figura 6 : RdP de unidades sencillas

Exclusión Mutua

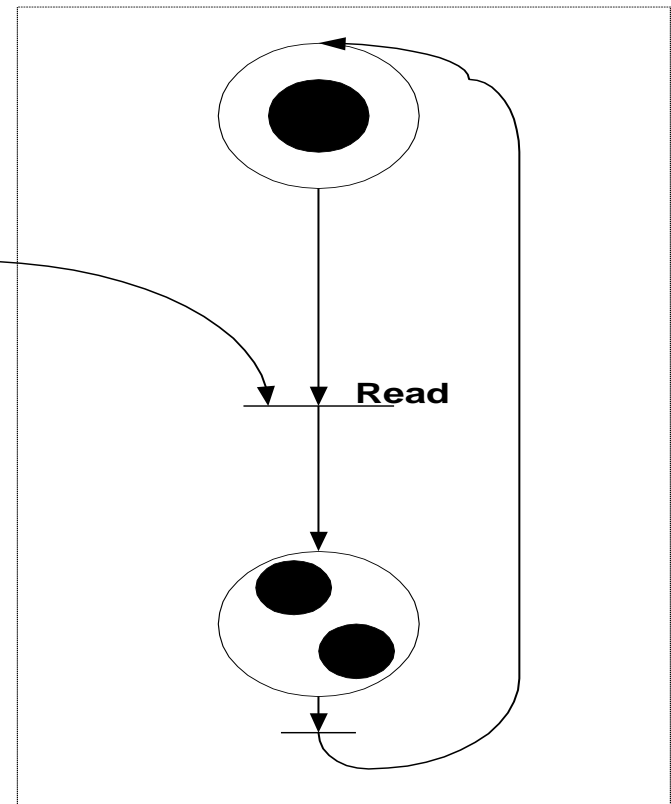


Producer - Consumidor

**Producer
processes**

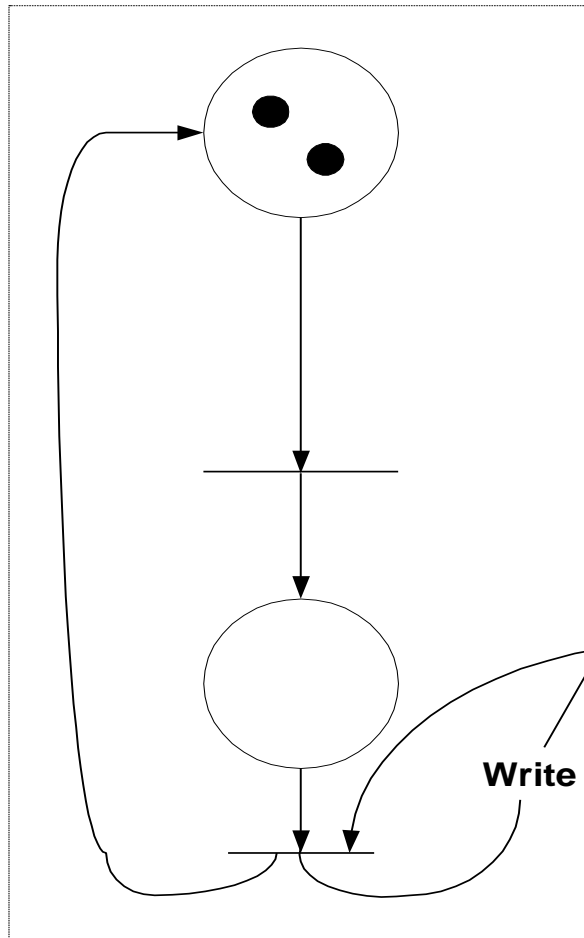


**Consumer
processes**

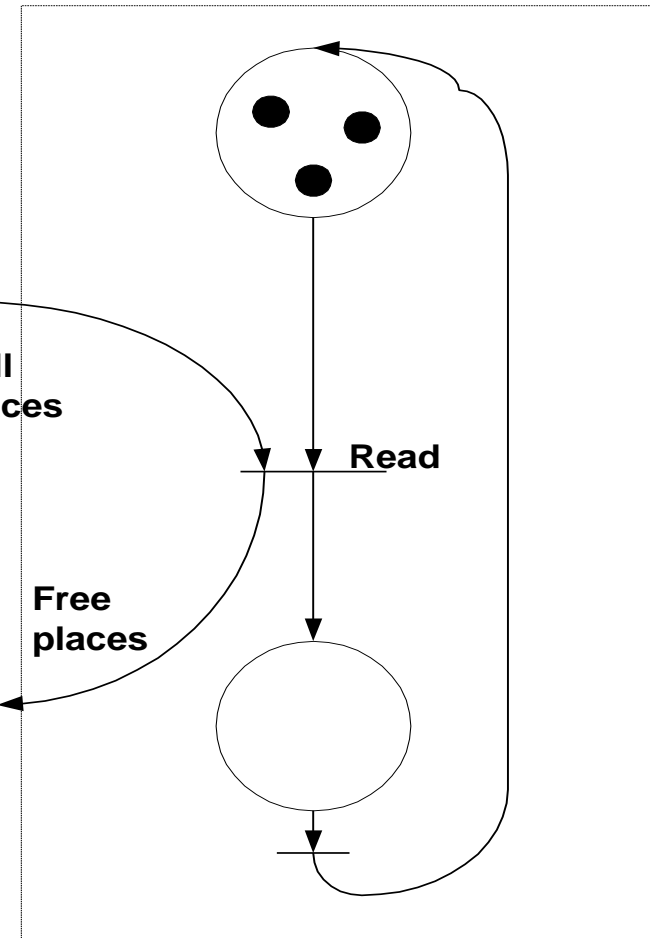


Buffer de destino:

**Producer
processes**



**Consumer
processes**



Buffer

**Full
places**

**Free
places**

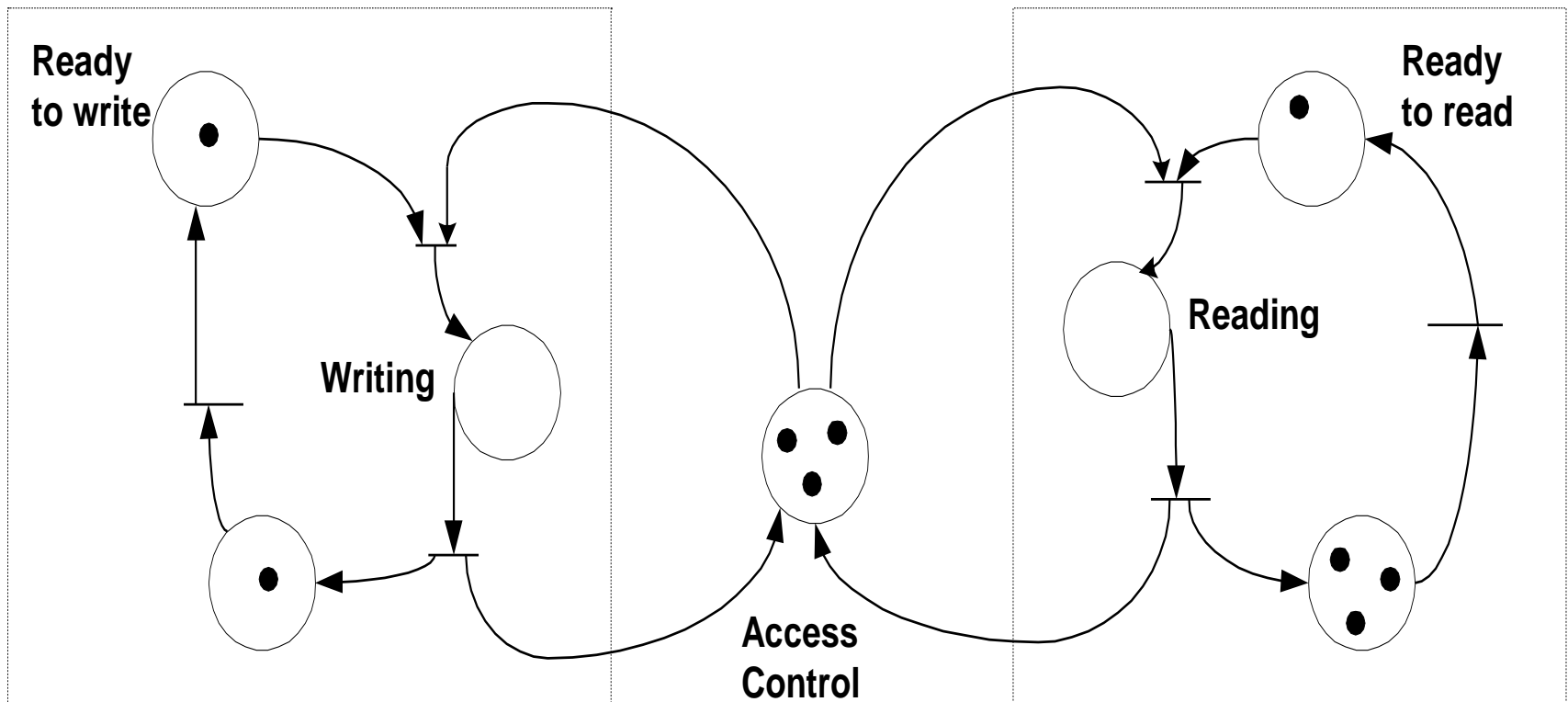
Write

Read

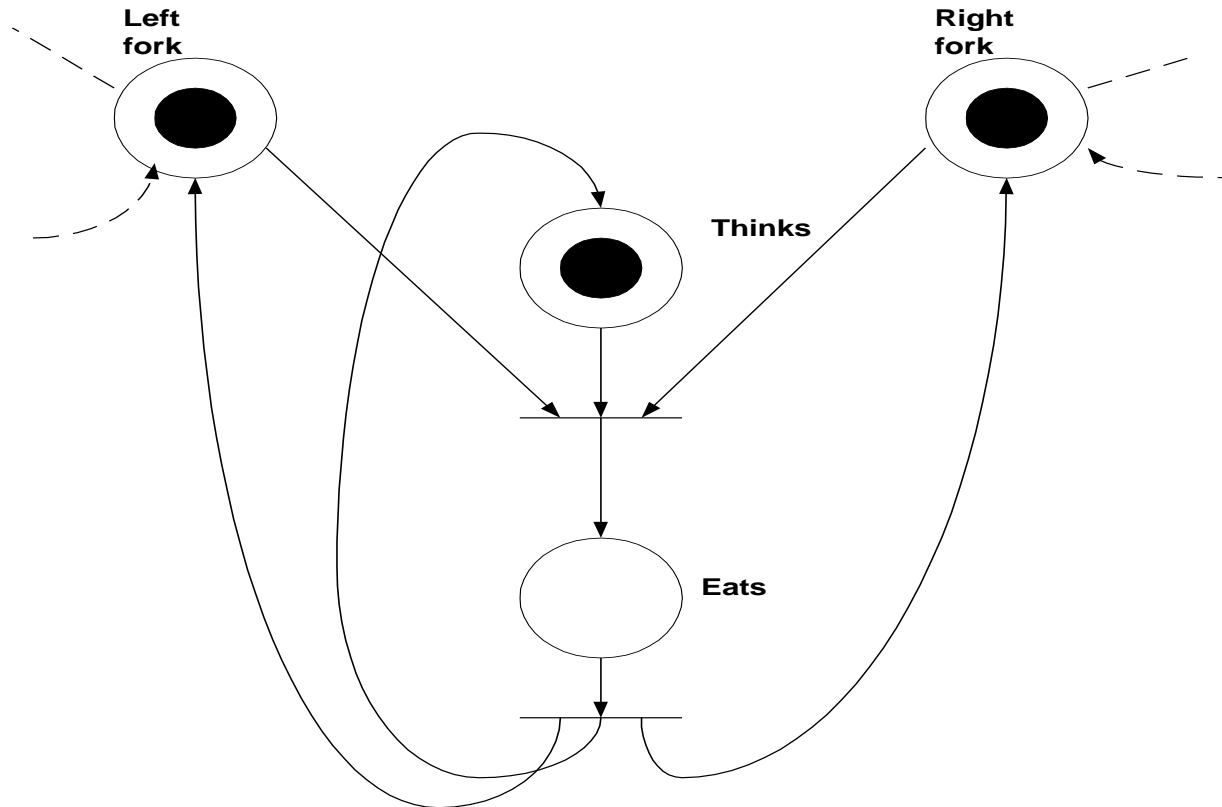
Lectores y escritores

Write processes

Readers processes

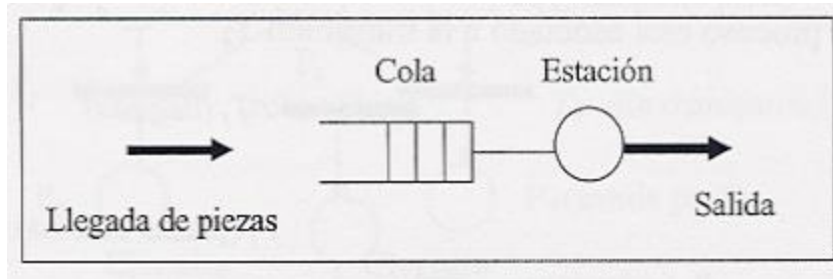


Filósofos comensales



Exemple MM2

- Estació de treball
 - ▣ Dos servidors
 - ▣ Una única cua



Xarxes de Petri MM2

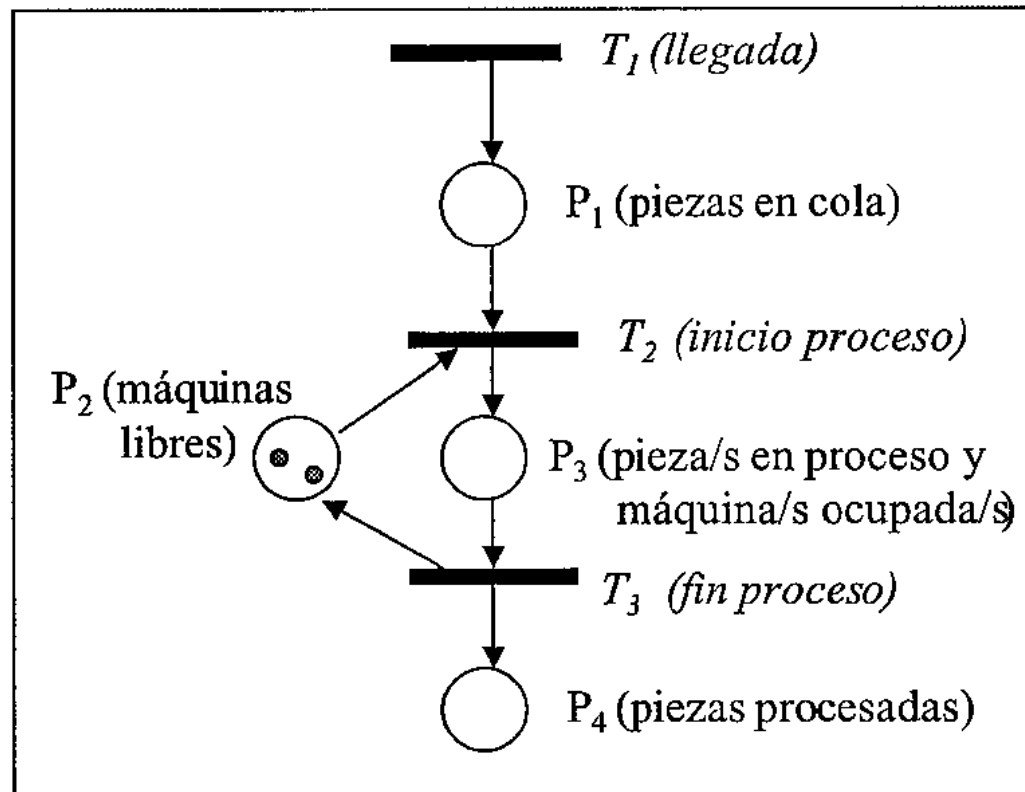
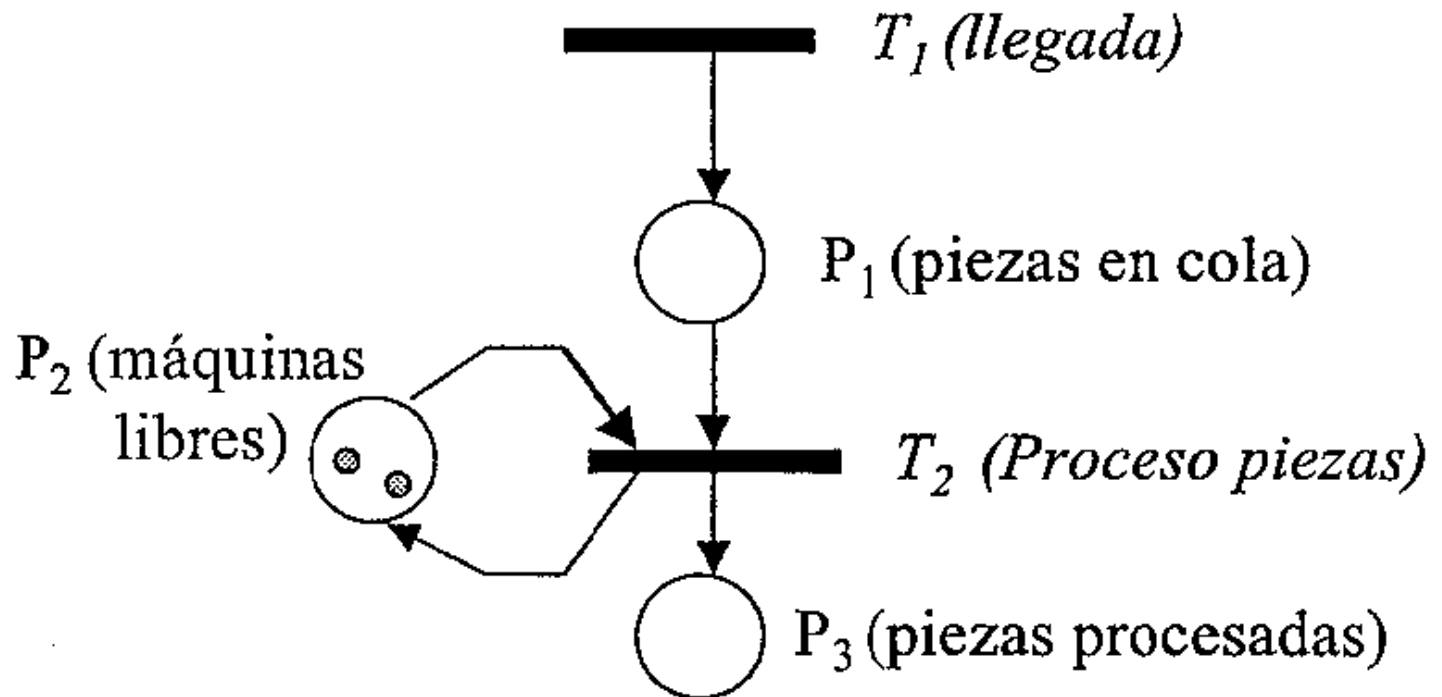


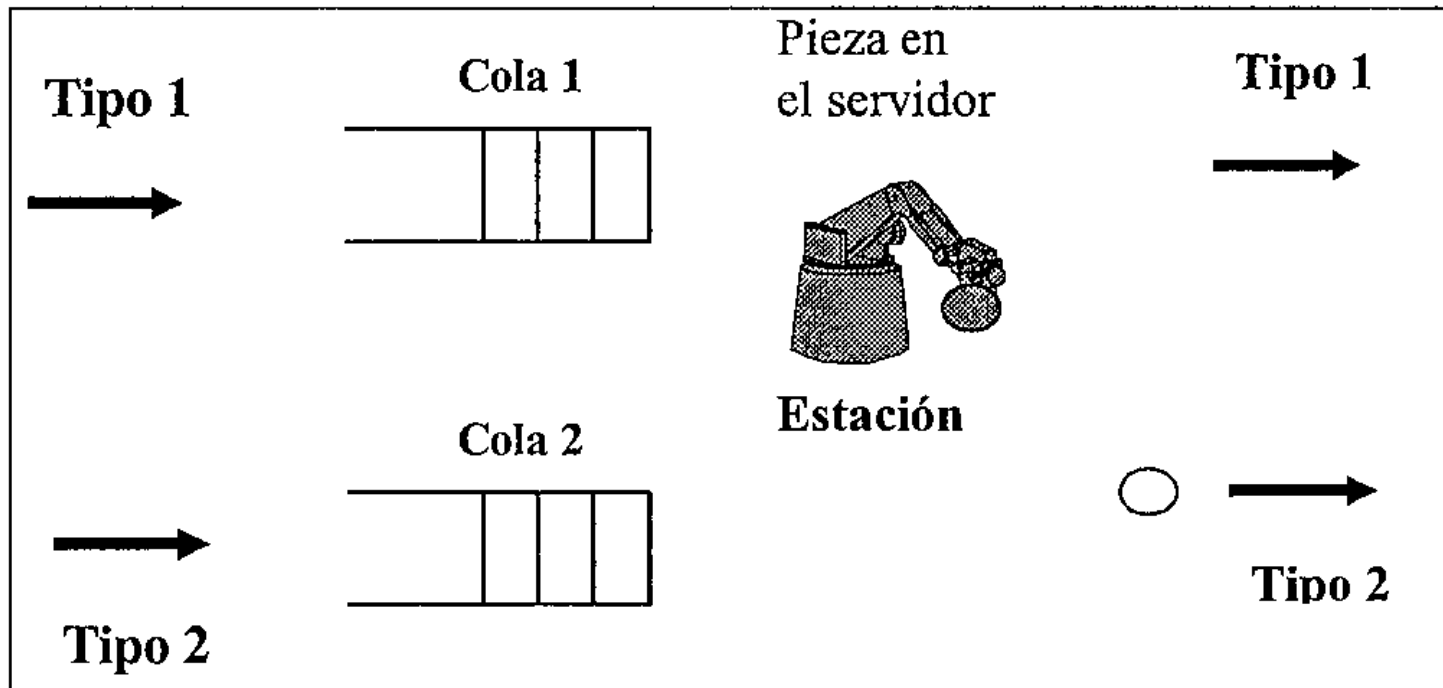
Figura 8. RdP de la estación de trabajo.

Xarxes de Petri

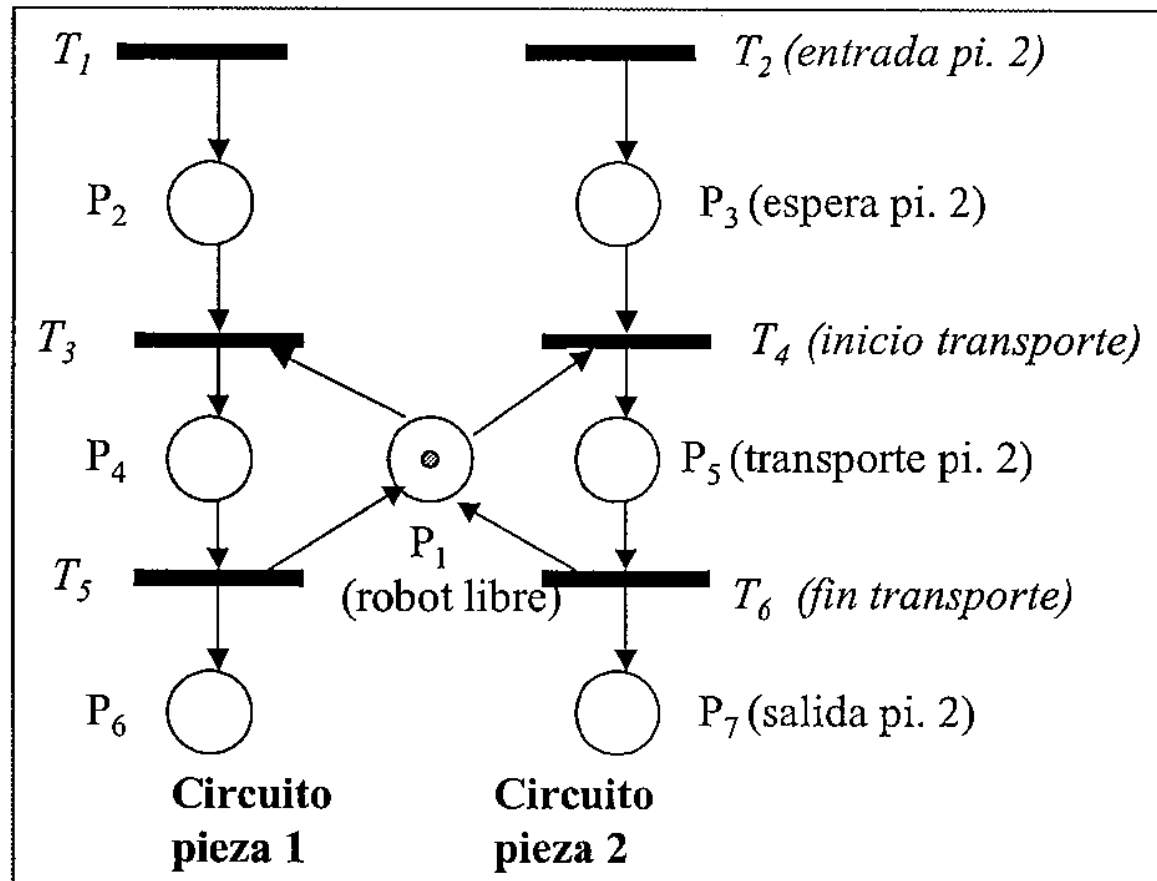
- No contemplen de forma explícita el temps consumit pels elements que recorren la xarxa.



Exemple Robot



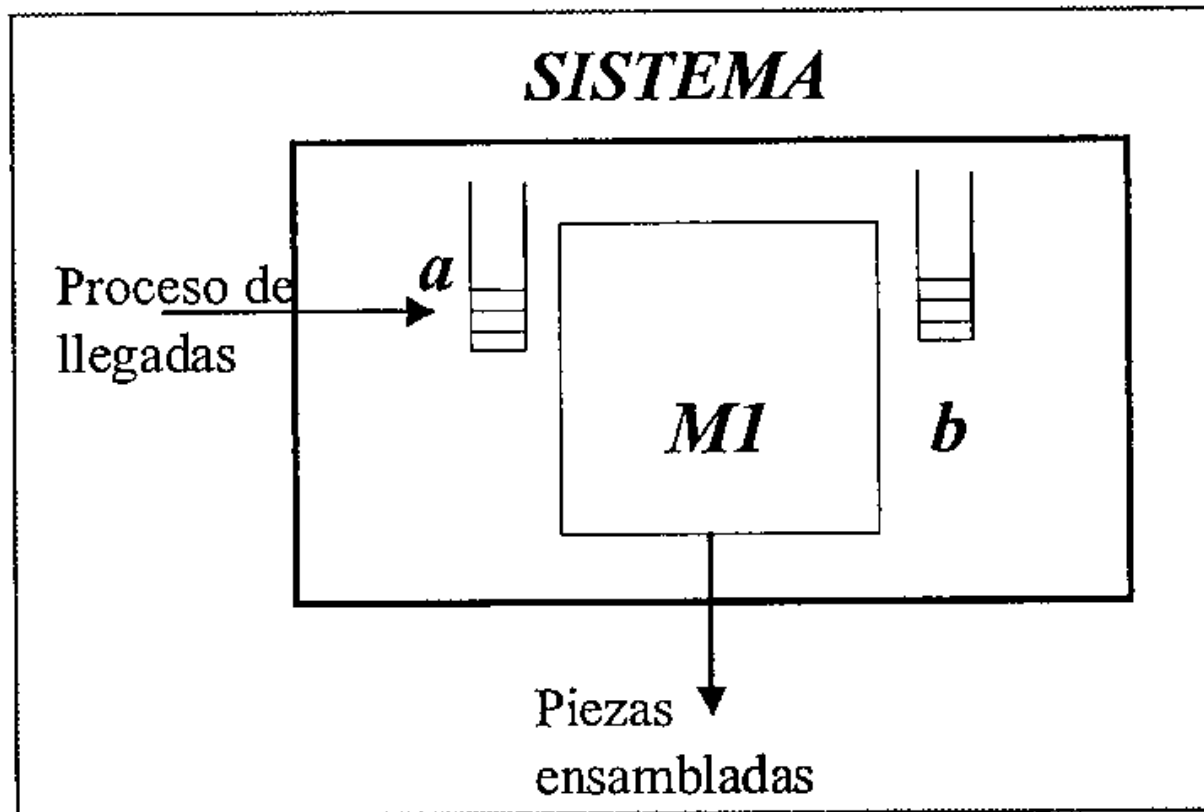
Exemple Robot



Exemple ASSEMBLE

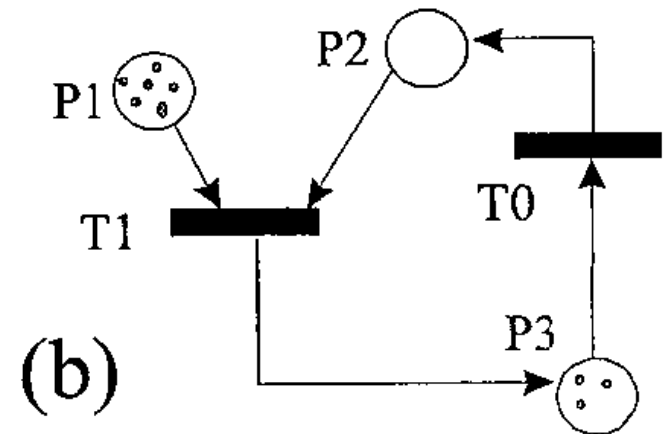
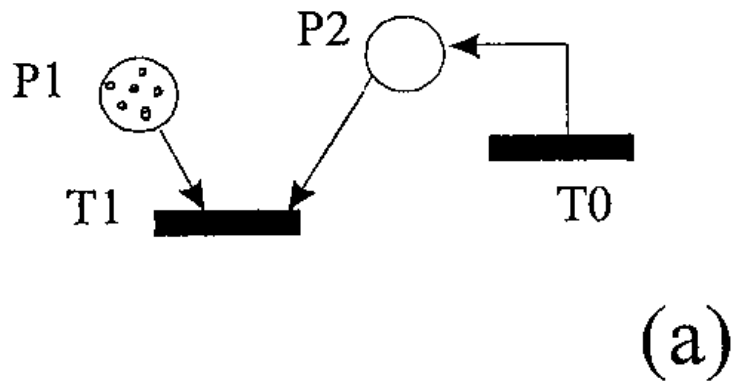
- Maquina d'assemblatge.
 - ▣ Dos tipus de peces A i B.
 - ▣ Cada cop unim una peça A amb una de B.
 - ▣ Tenim un nombre de peces A definit inicialment.
- Alternatives:
 1. Arriben peces de tipus B contínuament.
 2. Tenim únicament 3 peces B reaprofitables

Exemple ASSEMBLE



Exemple ASSEMBLE

- L'opció b suposa un límit de 3 unitats en l'estoc de peces B.



Xarxes de Petri

Anàlisi de les xarxes de Petri

Formalització de models de simulació

XdP: Anàlisi

- L'anàlisi sobre les xarxes de Petri permet:
 - ▣ Determinar els **bloquejos** del sistema.
 - ▣ Trobar possibles **camins per abastar** un **estat final** a partir de un estat inicial, i obtenir el cost de cada un d'aquests camins.
 - ▣ Obtenir un conjunt de **possibles estats** als que es pot arribar a partir de un estat inicial.

XdP: Arbre d'abast

- Donada una XdP $N=(P, T, A, W, M_0)$ es vol saber si:
 - ▣ Existeix una seqüència σ de transicions que permetin abastar un estat M a partir del estat inicial M_0
 - ▣ $M_0 \Rightarrow \sigma \Rightarrow M$
 - ▣ Funció d'abastabilitat $R(M_0)$
 - ▣ Es construeix un arbre d'abast.

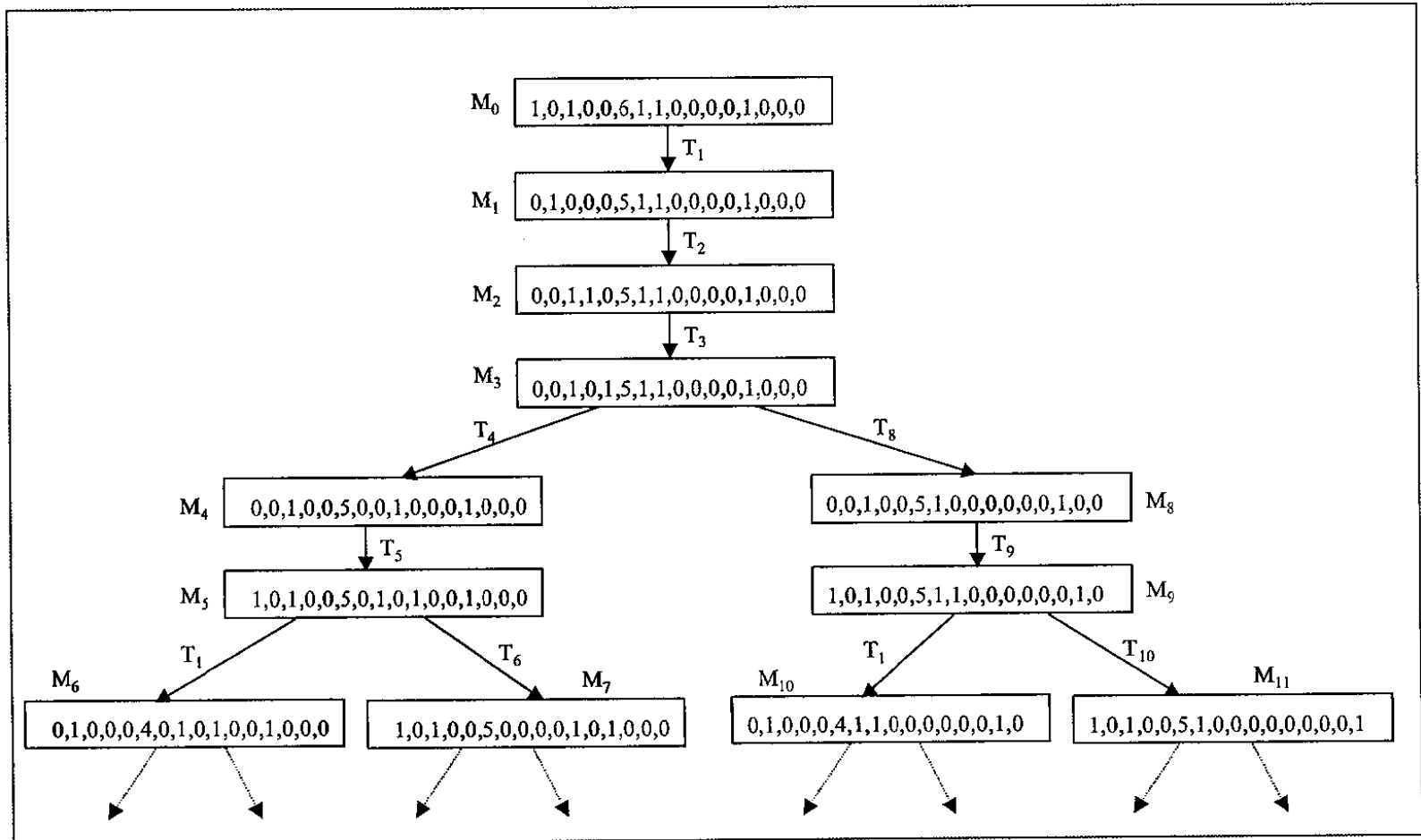
XdP: Construcció d'un arbre d'abast I

- L'arrel del arbre es el estat inicial del sistema (vector M_0)
- Per cada un dels nodes del arbre es generen tants nodes fills com transicions hi ha activades.
 - ▣ Els nodes filles representen el estat de la XdP una vegada s'ha disparat cada una de les transicions activades.
 - ▣ Els arcs que connecten els nodes pares amb els fills indiquen gràficament la transició que s'ha disparat.

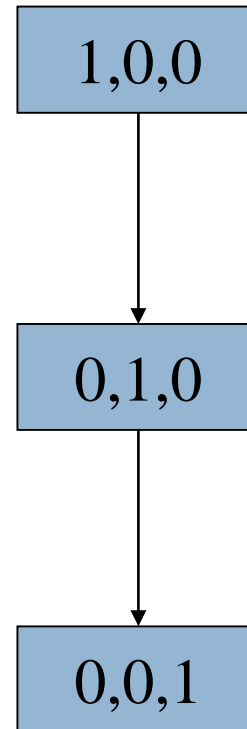
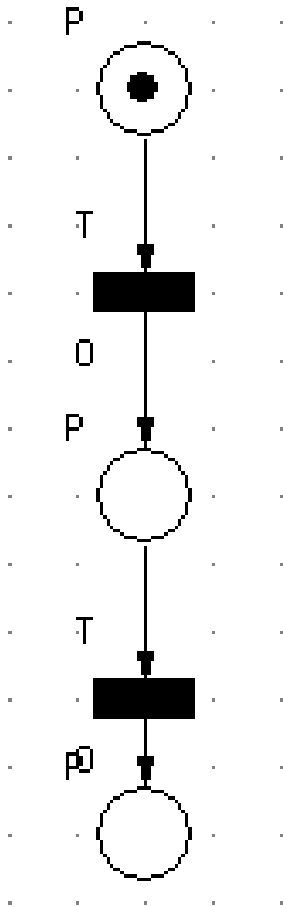
XdP: Construcció d'un arbre d'abast II

- Quan es genera un node fill que ja existeix en algun altre nivell del arbre, anterior o igual al actual, aquest node es marca com “**old**” i no es generen nous fills a partir d'ell.
- Un node del arbre que no te cap transició activada es marca com “**dead end**”
- Un node que no està marcat com “old” o “dead end” es un node “**new**”.

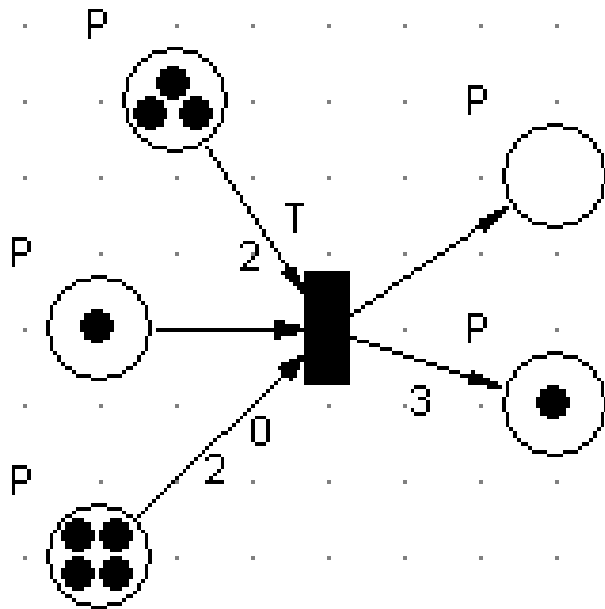
XdP: Exemple d'un arbre d'abast



XdP: Arbre d'abast



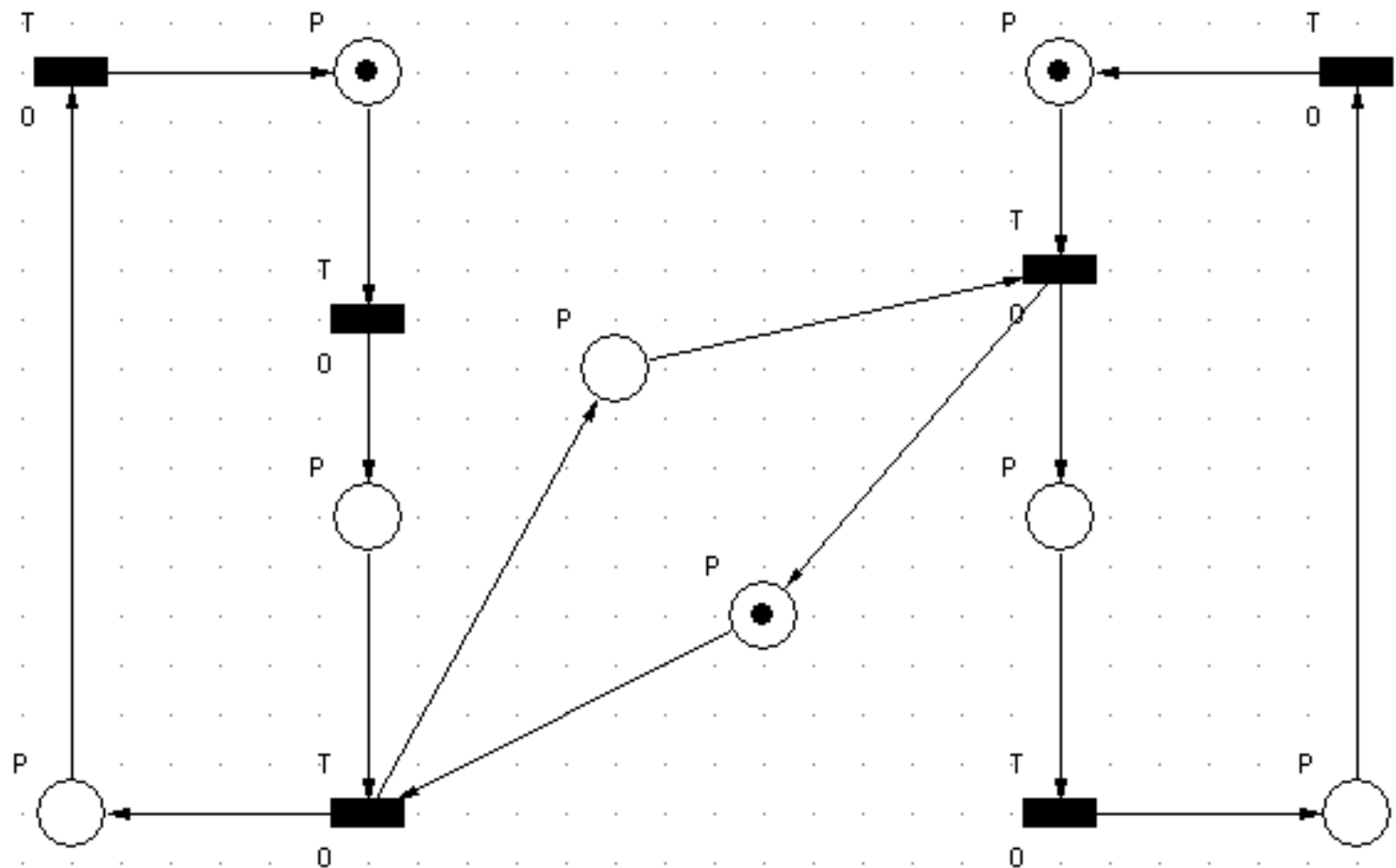
XdP: Arbre d'abast



3,1,4,0,1

1,0,2,1,4

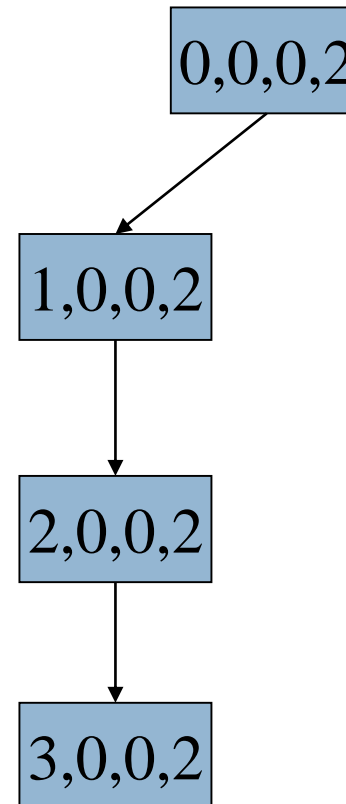
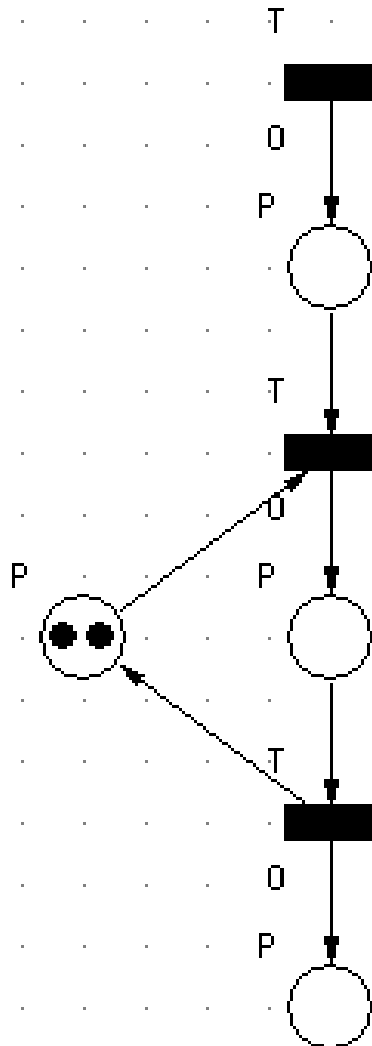
XdP: Productor-Consumidor



XdP: Productor consumidor

- [0]{1,0,0,0,1,1,0,0} [0]{0,1,0,0,1,1,0,0}
- [0]{0,1,0,0,1,1,0,0} [0]{0,0,1,1,0,1,0,0}
- [0]{0,0,1,1,0,1,0,0} [0]{1,0,0,1,0,1,0,0} [0]{0,0,1,0,1,0,1,0}
- [0]{0,0,1,0,1,0,1,0} [0]{1,0,0,0,1,0,1,0} [0]{0,0,1,0,1,0,0,1}
- [0]{1,0,0,1,0,1,0,0} [0]{0,1,0,1,0,1,0,0} [0]{1,0,0,0,1,0,1,0}
- [0]{0,0,1,0,1,0,0,1} [0]{1,0,0,0,1,0,0,1} [0]{0,0,1,0,1,1,0,0}
- [0]{1,0,0,0,1,0,1,0} [0]{0,1,0,0,1,0,1,0} [0]{1,0,0,0,1,0,0,1}
- [0]{0,1,0,1,0,1,0,0} [0]{0,1,0,0,1,0,1,0}
- [0]{0,0,1,0,1,1,0,0} [0]{1,0,0,0,1,1,0,0}
- [0]{1,0,0,0,1,0,0,1} [0]{0,1,0,0,1,0,0,1} [0]{1,0,0,0,1,1,0,0}
- [0]{0,1,0,0,1,0,1,0} [0]{0,0,1,1,0,0,1,0} [0]{0,1,0,0,1,0,0,1}
- [0]{0,1,0,0,1,0,0,1} [0]{0,0,1,1,0,0,0,1} [0]{0,1,0,0,1,1,0,0}
- [0]{0,0,1,1,0,0,1,0} [0]{1,0,0,1,0,0,1,0} [0]{0,0,1,1,0,0,0,1}
- [0]{0,0,1,1,0,0,0,1} [0]{1,0,0,1,0,0,0,1} [0]{0,0,1,1,0,1,0,0}
- [0]{1,0,0,1,0,0,1,0} [0]{0,1,0,1,0,0,1,0} [0]{1,0,0,1,0,0,0,1}
- [0]{1,0,0,1,0,0,0,1} [0]{0,1,0,1,0,0,0,1} [0]{1,0,0,1,0,1,0,0}
- [0]{0,1,0,1,0,0,1,0} [0]{0,1,0,1,0,0,0,1}
- [0]{0,1,0,1,0,0,0,1} [0]{0,1,0,1,0,1,0,0}
- Ordering:
- p0; p1; p2; p3; p4; p5; p6; p7

XdP: Arbre d'abast MM2



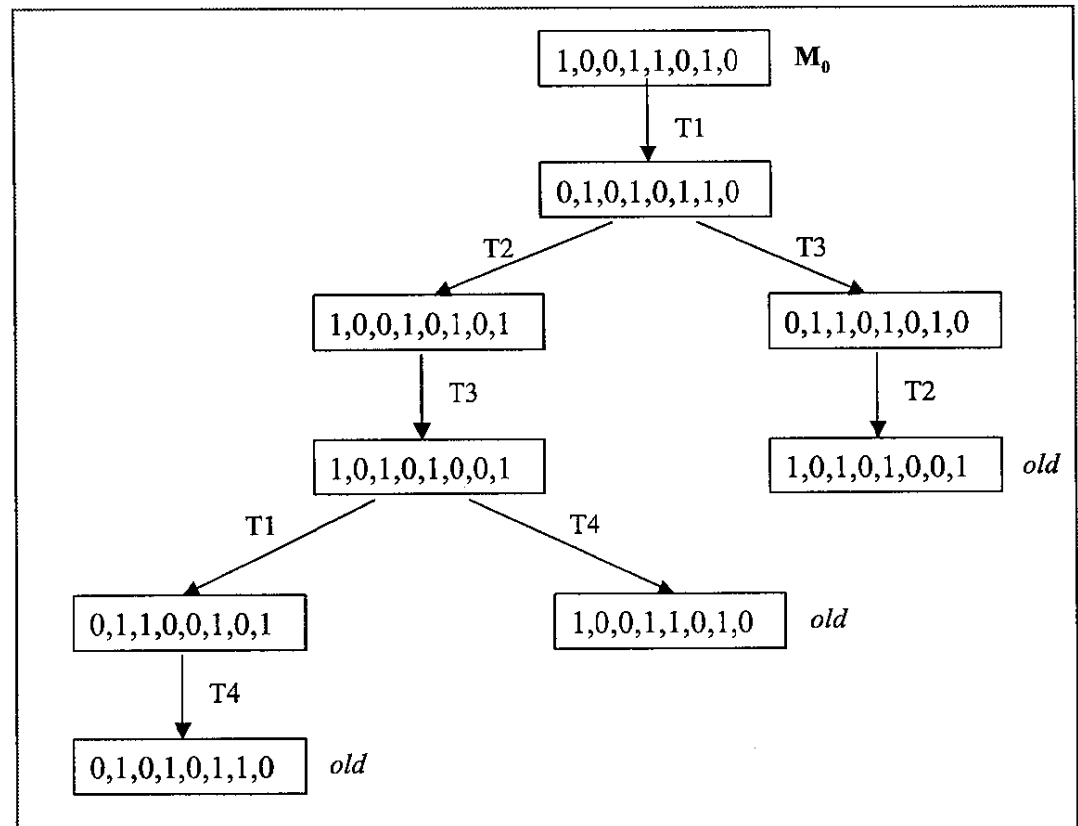
No es finit!

XdP: Acotació

- Serveix per estudiar si la XdP creix fins al infinit.
 - ▣ Un lloc $P_i \in P$ en una XdP N amb estat inicial M_0 anomena k -acotat si $M(P_i) \leq k$ per tots els estats $M \in R(M_0)$ i k es enter diferent de infinit.
 - ▣ L'arbre d'abast permet determinar si la XdP està acotada.

XdP: Exemple Acotació

- Arbre 1-acotat (tots els llocs son 1-acotats)



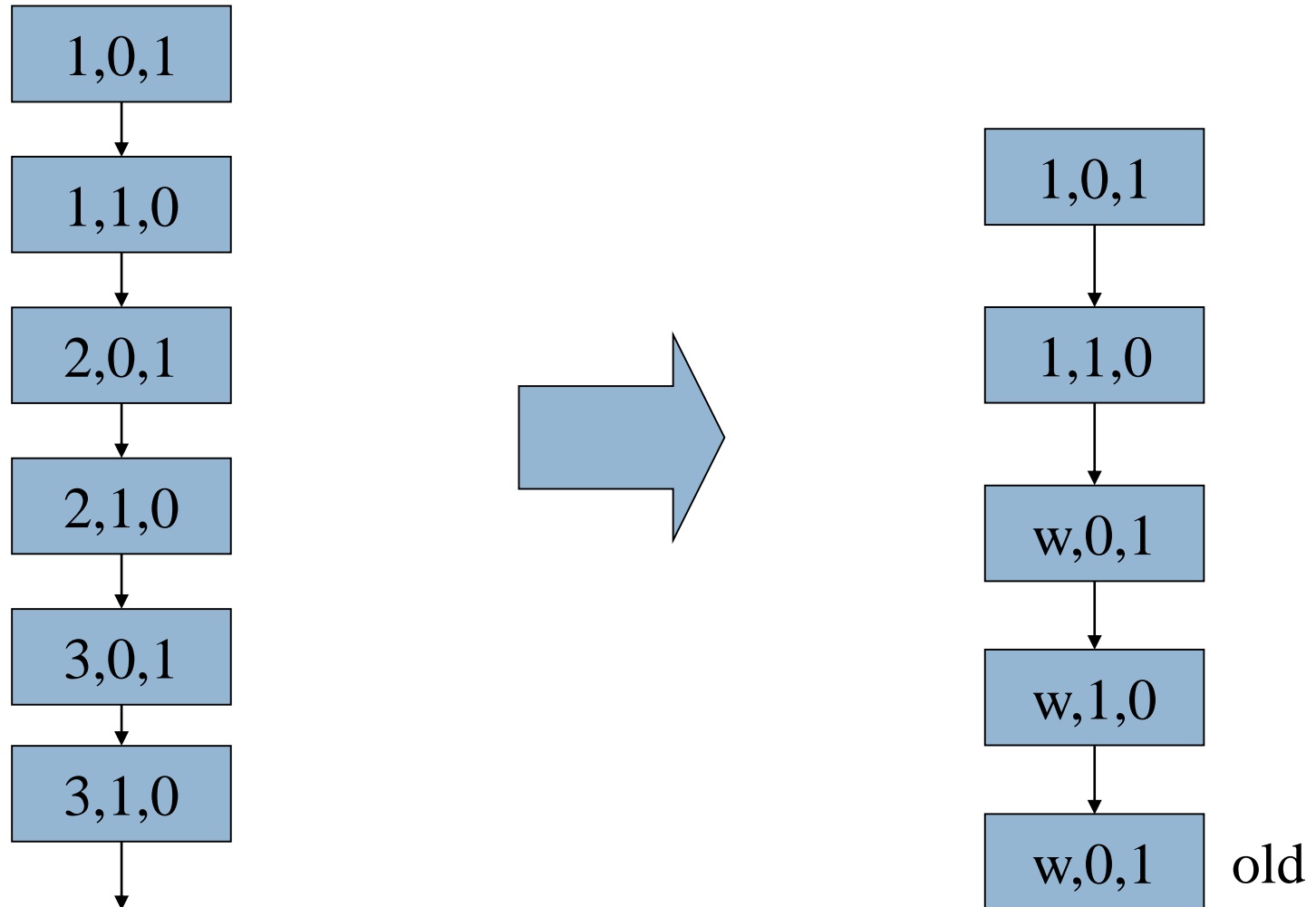
XdP: Arbre de Cobertura I

- S'usa quan la XdP no està acotada.
- S'afegeix la següent regla addicional a la construcció d'un arbre d'abast.
 - ▣ Si existeix un node M en el camí que hi ha entre el node arrel M_0 i un node M^* de qualsevol nivell del arbre tal que:
 - $M^*(p) \geq M(p) \quad \forall p_i \in P$
 - $M^*(p_k) > M(p_k)$ per algun $p_k \in P$ com a mínim

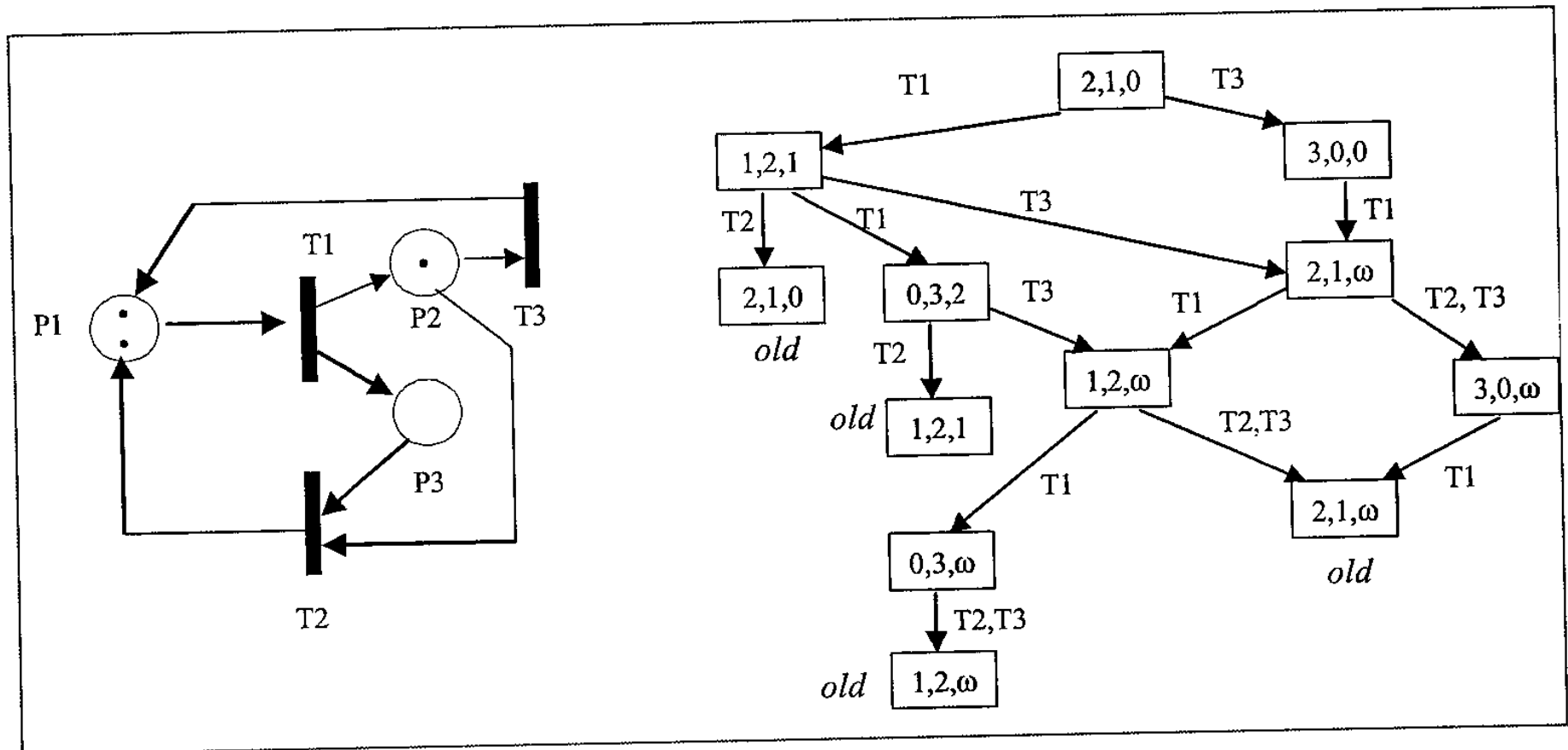
XdP: Arbre de Cobertura II

- El valor p_k es substitueix per “w”, on “w” significa infinit.
- El valor de p_k es mantindrà w en els fills successius del node M^* .
- Tot el conjunt de regles es també vàlid per als nodes amb valors w.

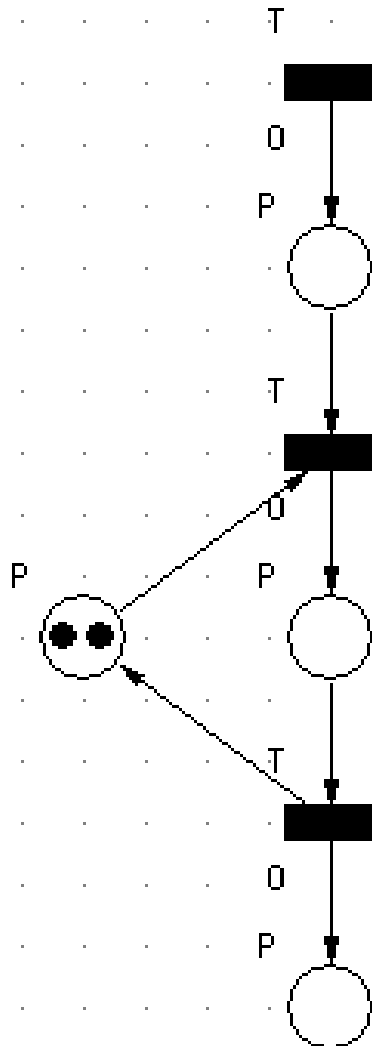
Exemple Arbre de cobertura



XdP: Exemple d'arbre de cobertura

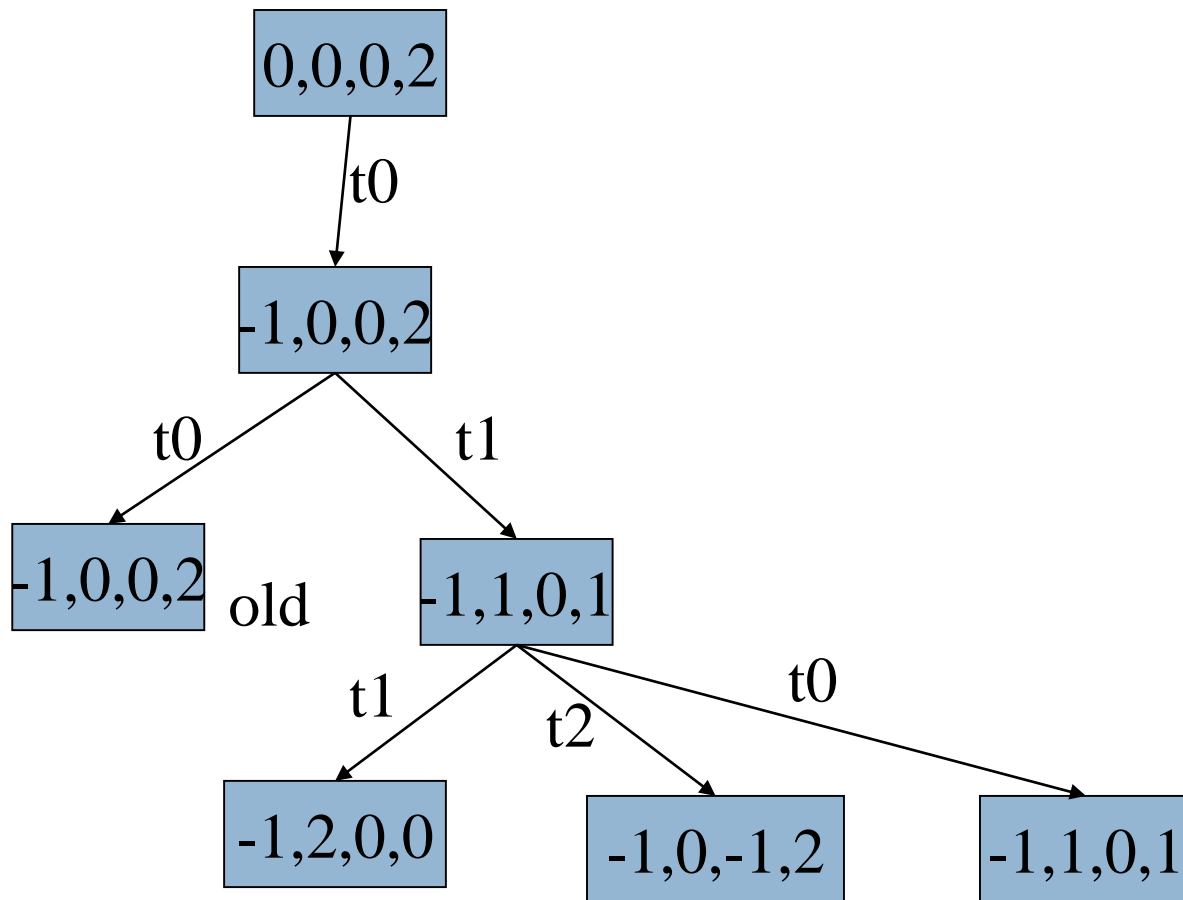


XdP: Arbre de cobertura MM2



- Comencem construint l'arbre d'abast.

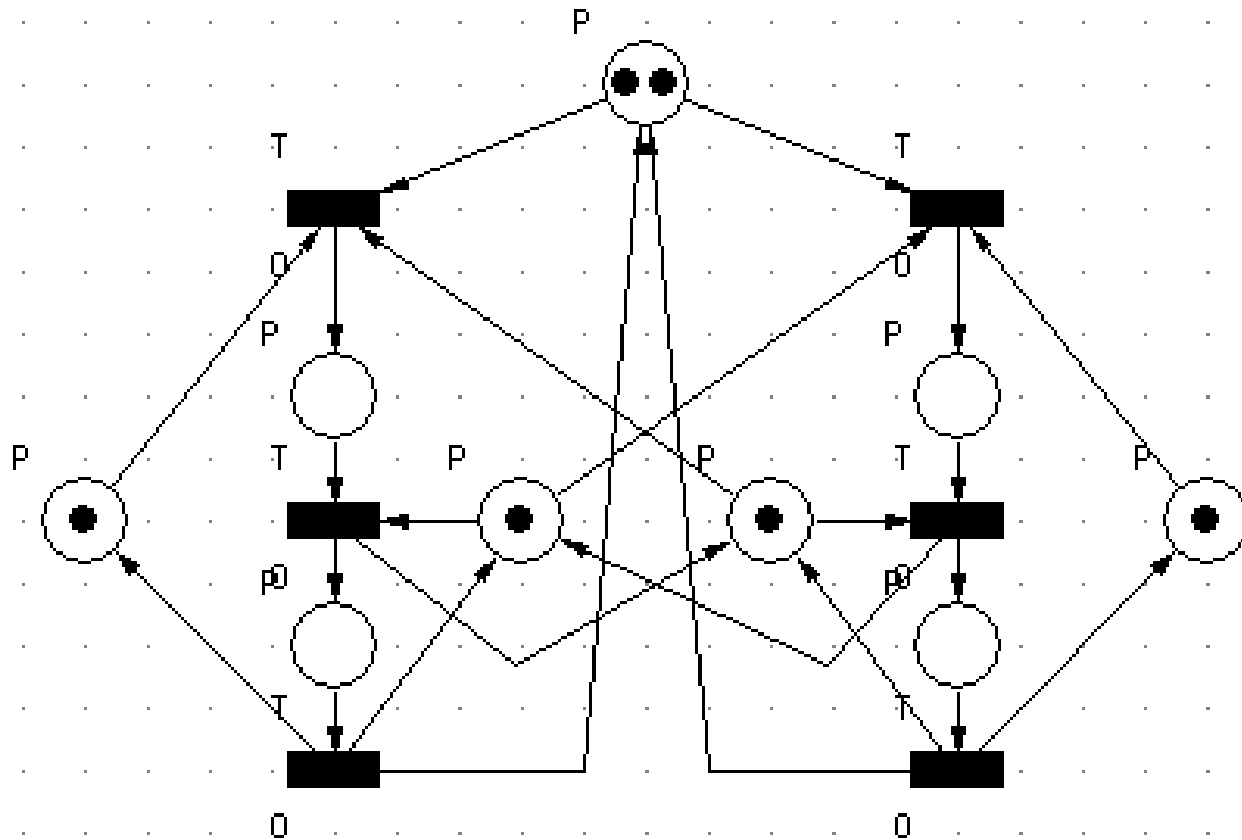
XdP: Arbre de couverture MM2



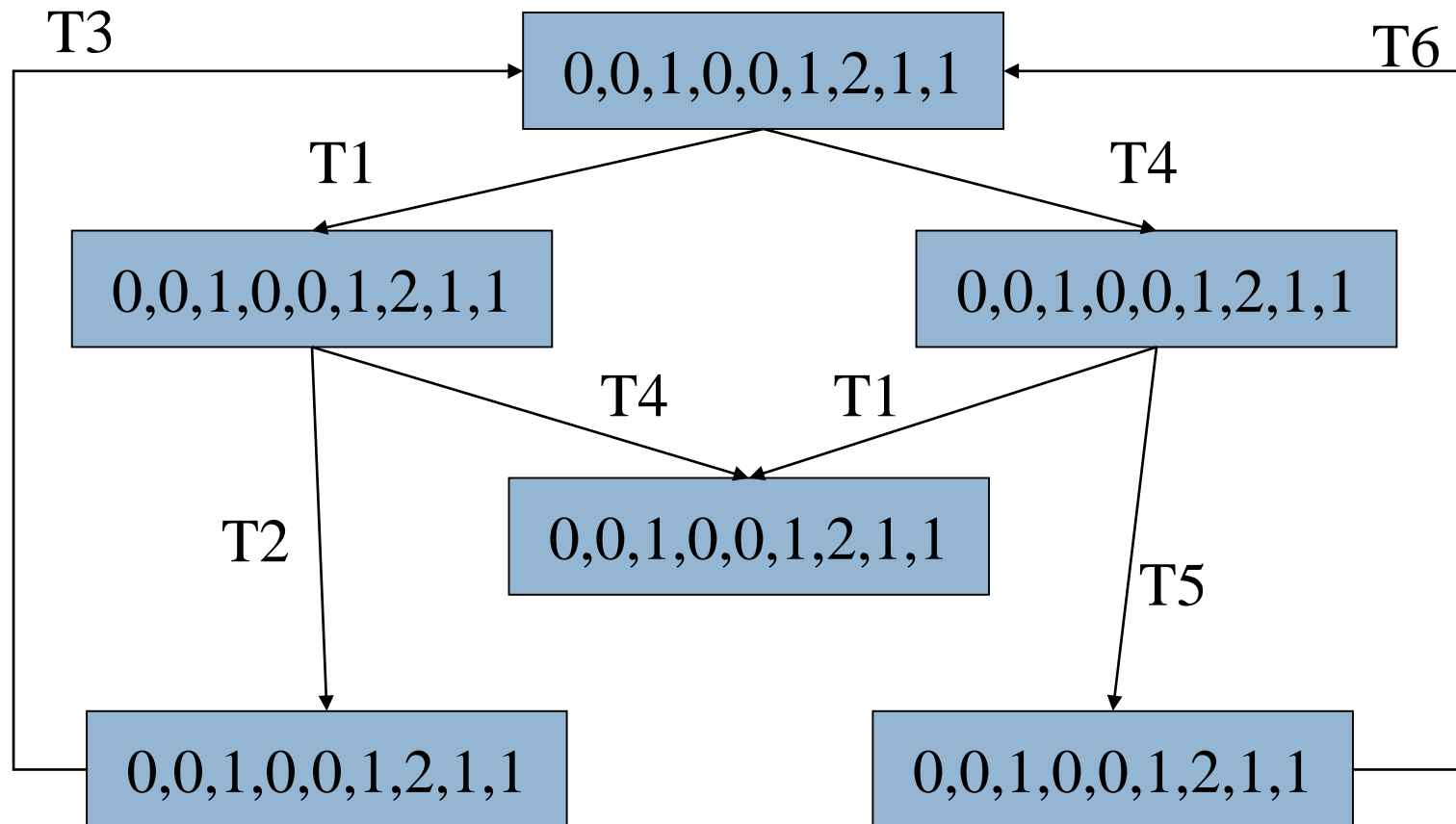
Llista d'estats

- $[5]\{0,0,0,2\} [4]\{-1,0,0,2\}$
- $[4]\{-1,0,0,2\} [3]\{-1,1,0,1\} [4]\{-1,0,0,2\}$
- $[3]\{-1,1,0,1\} [2]\{-1,2,0,0\} [0]\{-1,0,-1,2\} [3]\{-1,1,0,1\}$
- $[0]\{-1,0,-1,2\} [0]\{-1,1,-1,1\} [0]\{-1,0,-1,2\}$
- $[2]\{-1,2,0,0\} [1]\{-1,1,-1,1\} [2]\{-1,2,0,0\}$
- $[0]\{-1,1,-1,1\} [0]\{-1,2,-1,0\} [0]\{-1,0,-1,2\} [0]\{-1,1,-1,1\}$
- $[1]\{-1,1,-1,1\} [0]\{-1,2,-1,0\} [0]\{-1,0,-1,2\} [0]\{-1,1,-1,1\}$
- $[0]\{-1,2,-1,0\} [0]\{-1,1,-1,1\} [0]\{-1,2,-1,0\}$
- Ordering:
- $p0; p1; p2; p3$

Exemple arbre de cobertura



Exemple d'arbre de cobertura



Xarxes de Petri

Xarxes de Petri acolorides

Xarxes de Petri temporals

Xarxes de Petri estocàstiques

Xarxes de Petri amb cues: QPN

Xarxes de Petri acolorides

XdPC

XdPC: Motivació

- Es important que les XdP mantinguin un nivell d'abstracció similar a la descripció de esdeveniments en el entorn de simulació en el que es codificarà el model.
 - ▣ Ajuda a la codificació.
 - ▣ Ajuda al manteniment.

Xarxes de Petri acolorides

- Permeten modelar el flux de informació agregada a les entitats.
 - ▣ Atributs de les entitats.
 - ▣ Varien a partir dels esdeveniments.
- Permeten construir models mes compactes i paramètrics.
 - ▣ Faciliten el manteniment.
 - ▣ Faciliten la posterior codificació.

XdPC

- Permeten associar a cada marca un valor de un conjunt donat (color).
- NO augmenten la potència de càlcul.

XdPC: Definició I

- $RdPC = (\Sigma, P, T, A, N, C, G, E, I)$ on
- $\Sigma = \{C_1, C_2, \dots, C_{nc}\}$ Conjunts finits i no buits de colors.
Permeten definir els atributs que s'han d'especificar en cada tipus d'entitat a modelar.
 - ▣ C_i : Atributs de l'entitat de tipus i .

XdPC: Definició II

- $P = \{P_1, P_2, \dots, P_{nc}\}$ Conjunt finit de nodes de lloc que permeten especificar l'estat del sistema.
- $T = \{T_1, T_2, \dots, T_{nc}\}$ Conjunt finit de nodes transició. Les transicions en el model de simulació corresponen a esdeveniments que es solen codificar con el inici o el final de una certa activitat, o bé amb la fi d'un succés extern, com seria el cas de un procés d'arribades.

XdPC: Definició III

- $A = \{A_1, A_2, \dots, A_{nc}\}$ Conjunt finit d'arcs.
- N: Funció de node, $N(A_i)$, que permet associar a cada arc els seus nodes terminals (el node origen i el node destí). Els nodes tenen que ser de diferent tipus, per tant, si un node es una transició l'altre ha de ser un lloc i al contrari.

XdPC: Definició IV

- C: Conjunt de funcions de color, $C(P_i)$, que permeten especificar per cada node lloc el tipus d'entitats que poden emmagatzemar-se.
 - ▣ $C(P_i) = C_i \quad P_i \in P \quad C_i \in \Sigma$
- G: Funció guarda associada als nodes de tipus transició, $G(T_i)$, que sol emprar-se per desinhibir l'esdeveniment associat a la transició en funció dels valors dels atributs de l'entitat a processar.

XdPC: Definició V

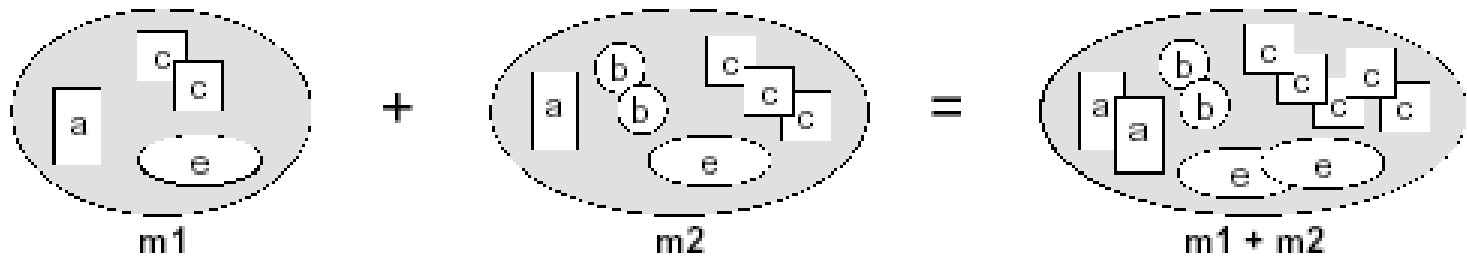
- E: Expressió associada a arcs, $E(A_i)$, que permeten especificar el tipus d'entitat del node lloc d'entrada a la transició que ha de recollir-se de entre les entitats emmagatzemades en el node lloc per habilitar el esdeveniment. Quan l'expressió E es troba associada a un arc de sortida de la transició, l'expressió s'empra per avaluar els nous valors dels atributs de les entitats de sortida.

XdPC: Definició VI

- l: Funció d'inicialització, $l(P_i)$, que permet especificar els valors dels colors (atributs) de les entitats inicialment emmagatzemades en el nodes de lloc.
 $8(1)+3(2)$

XdPC: Operacions (1/2)

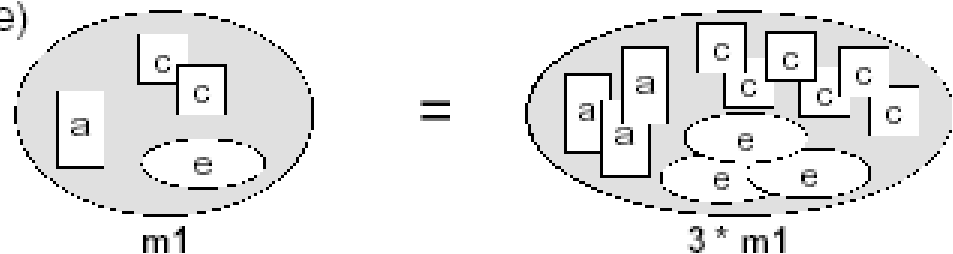
Addition (element-wise)



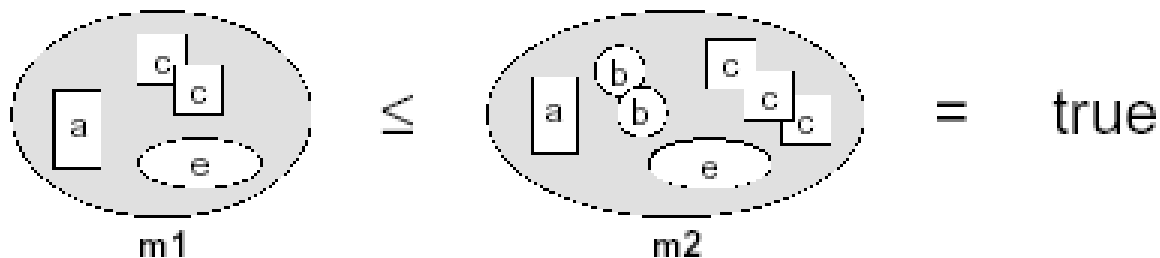
Scalar multiplication (element-wise)

3

*

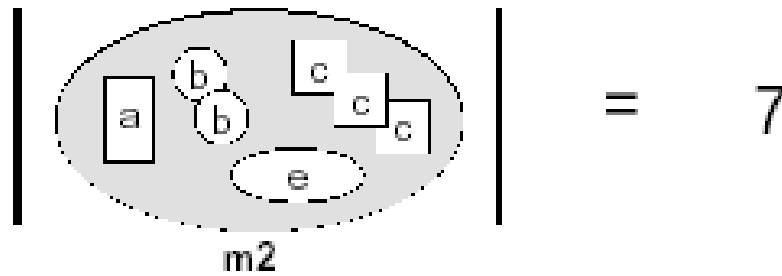


Comparison (element-wise)

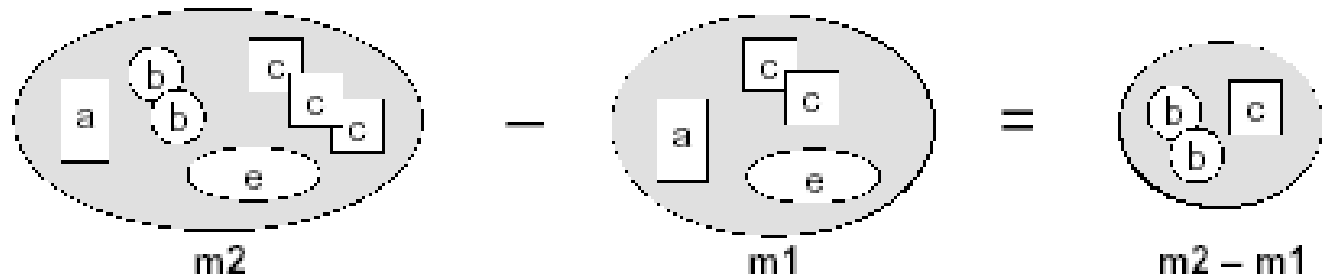


XdPC: Operacions (1 / 2)

Size (number of elements)



Subtraction (only if $m_2 \geq m_1$)

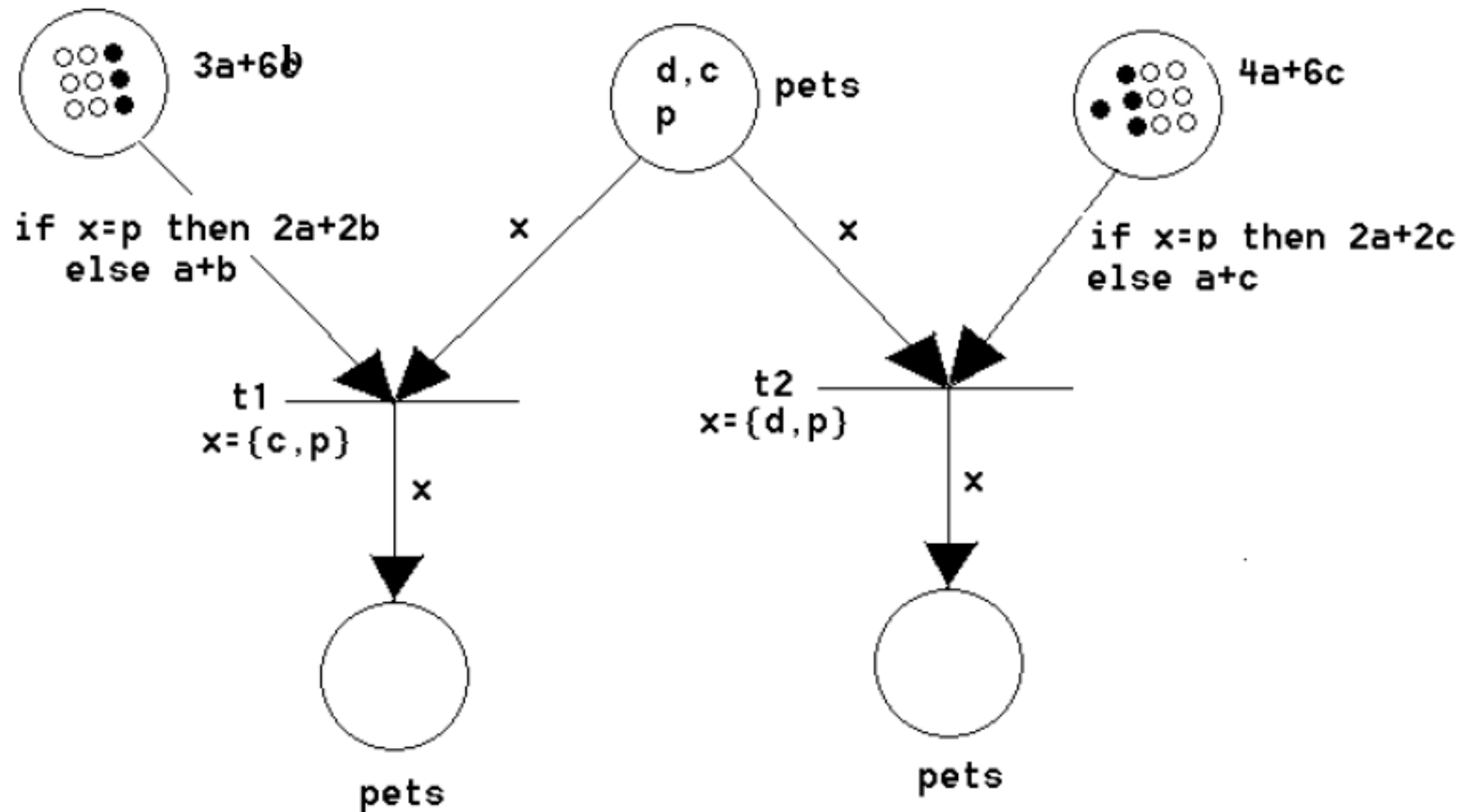


XdPC: Característiques

- Permeten establir en els arcs:
 - ▣ Condicions
 - ▣ Variables
- Colors (tipus de dades)
- Els llocs contenen marques de diferent color

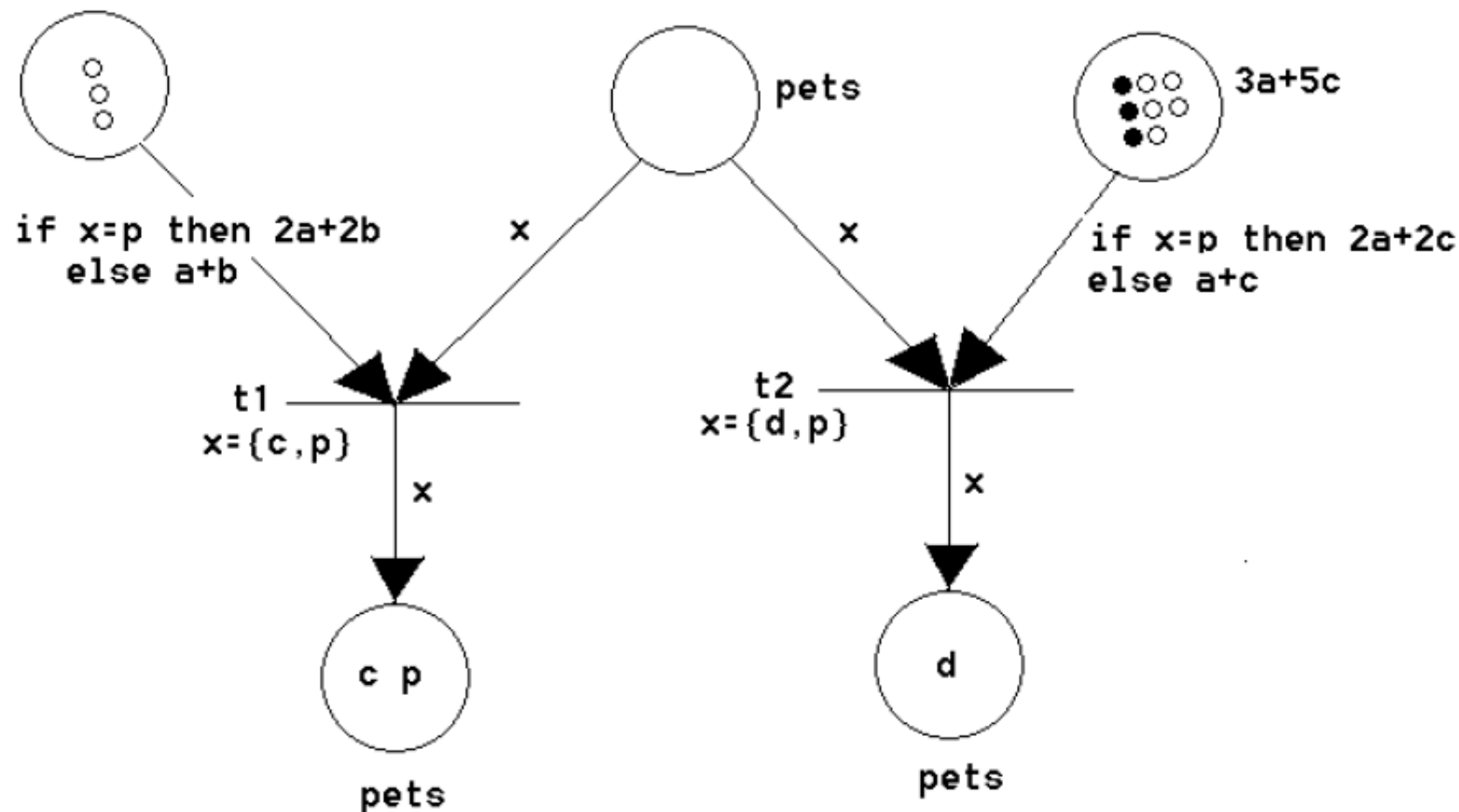
Example

Color cat_food=with b;
Color dog_food=with c;
Color pets=with dog| cat| pig;
Var x:pets;

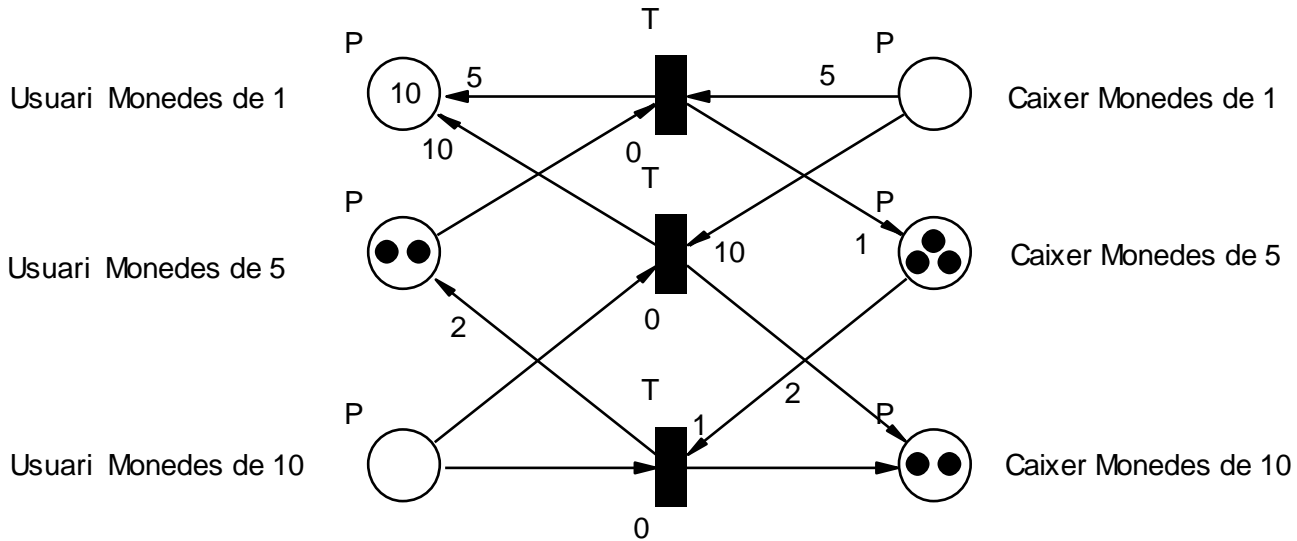


After firing occurrence

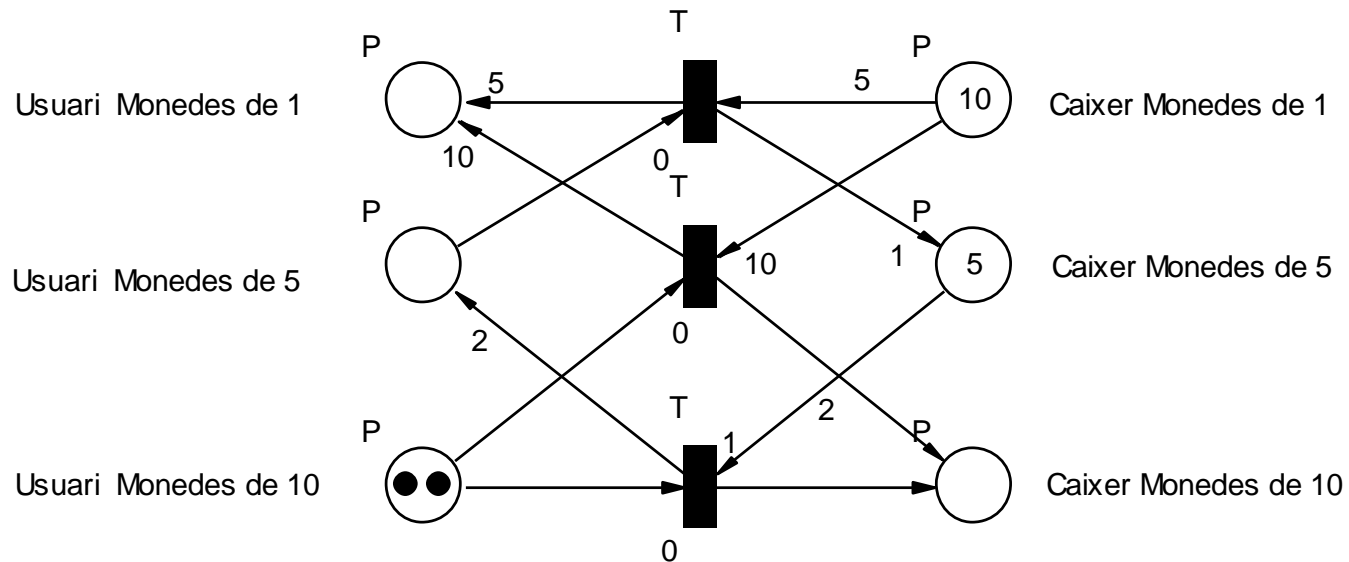
$$(t1, x=c) + (t1, x=p) + (t2, x=d)$$



XdPC: Exemple canvi

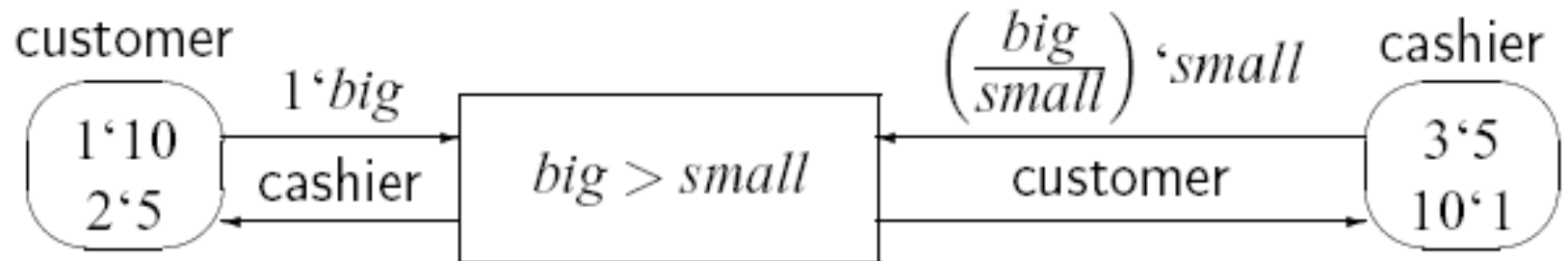


XdPC: Exemple canvi

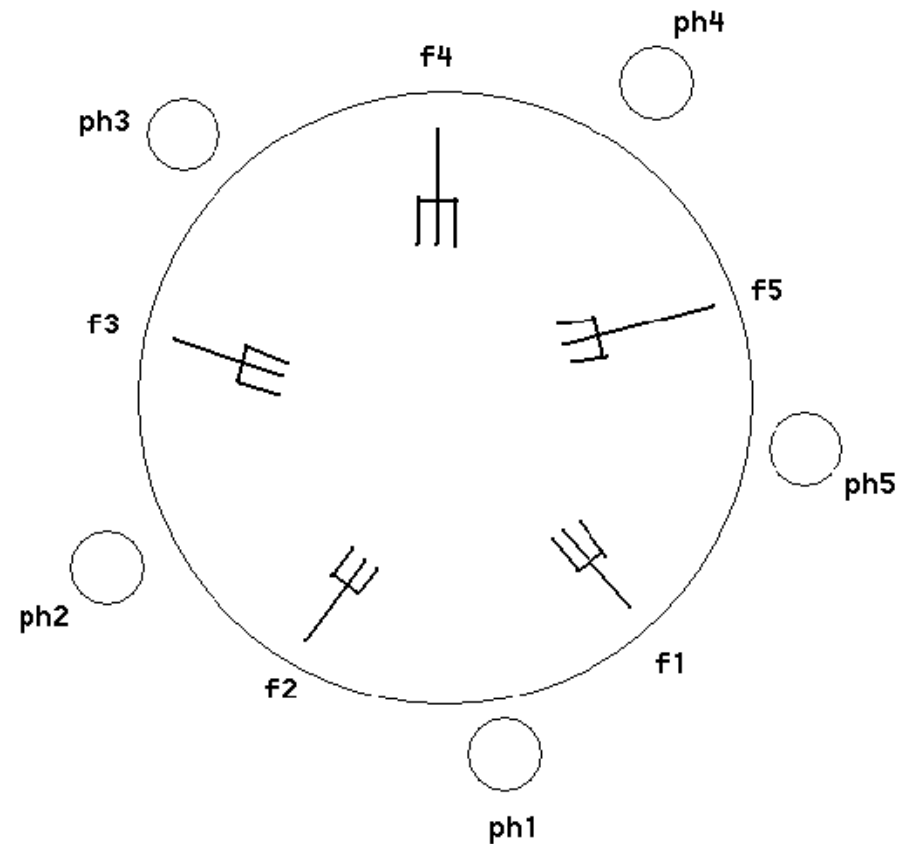


XdPC: Exemple canvi

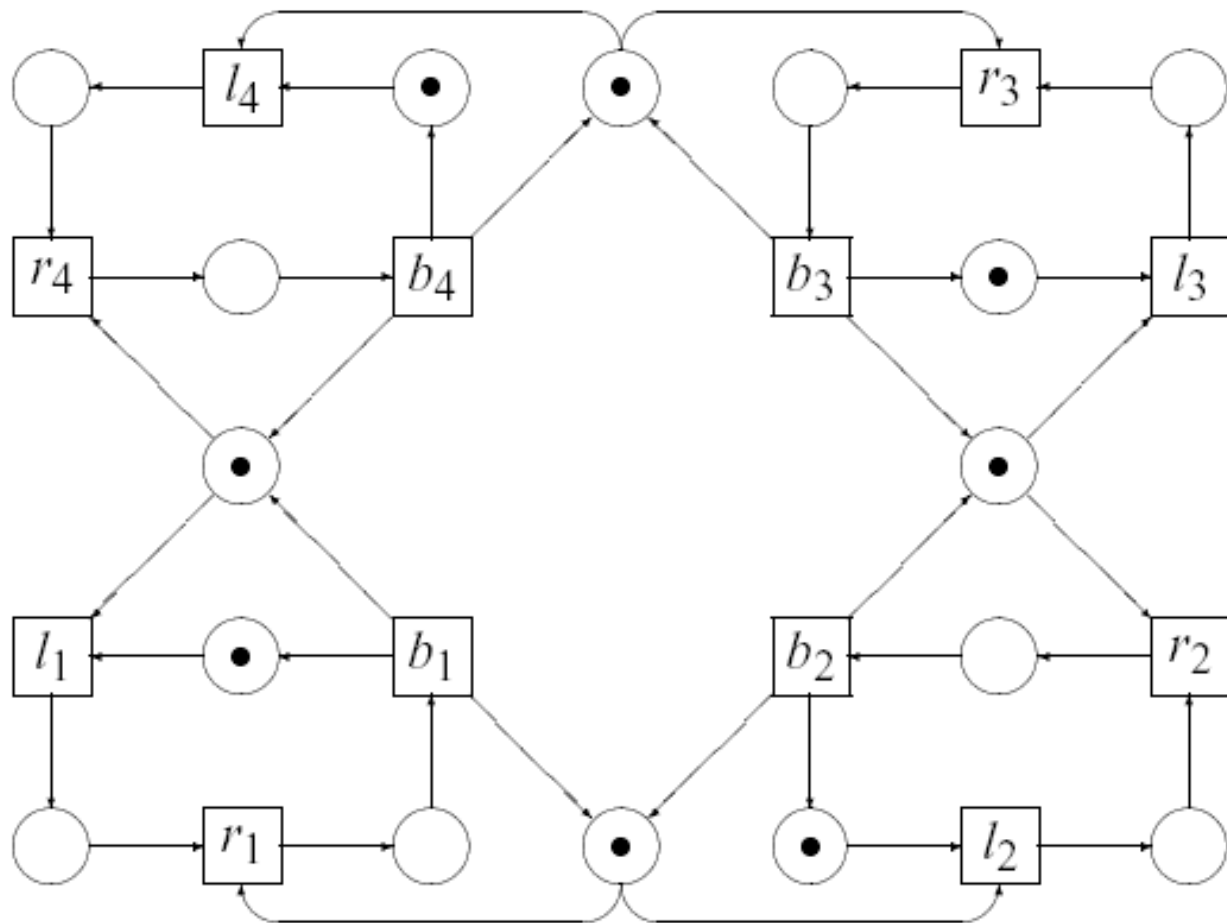
- Arbre de cobertura Xarxa de petri normal:
 - ▣ $[5,1,1,5,4,0], [10,0,1,0,5,0], [0,4,0,10,1,1], [5,3,0,5,2,1], [10,2,0,0,3,1]$
- XdPC equivalent:



XdPC: 5 filòsofs



XdPC: 5 filòsofs



colour PH=with ph1|ph2|ph3|ph4|ph5

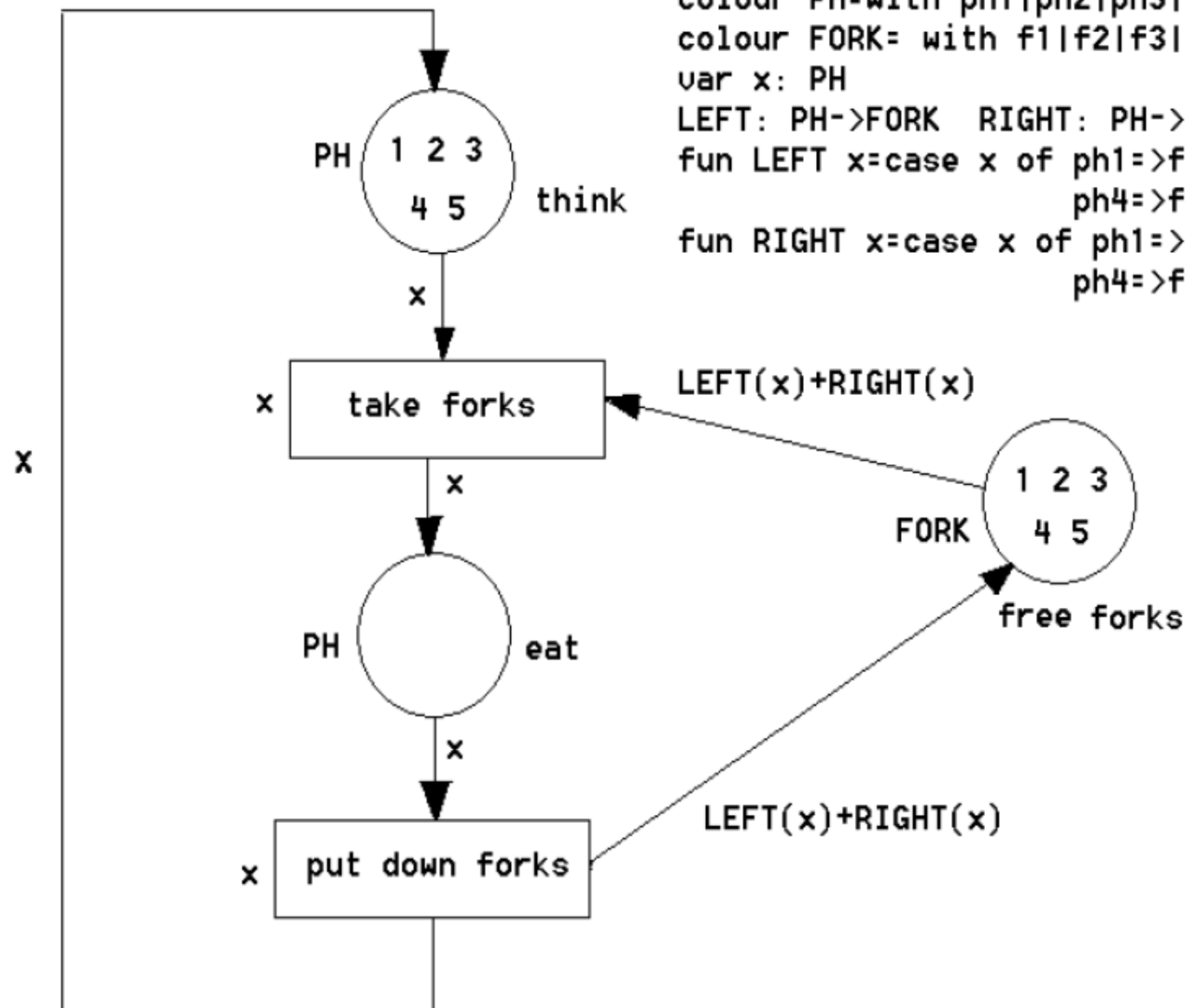
colour FORK= with f1|f2|f3|f4|f5

var x: PH

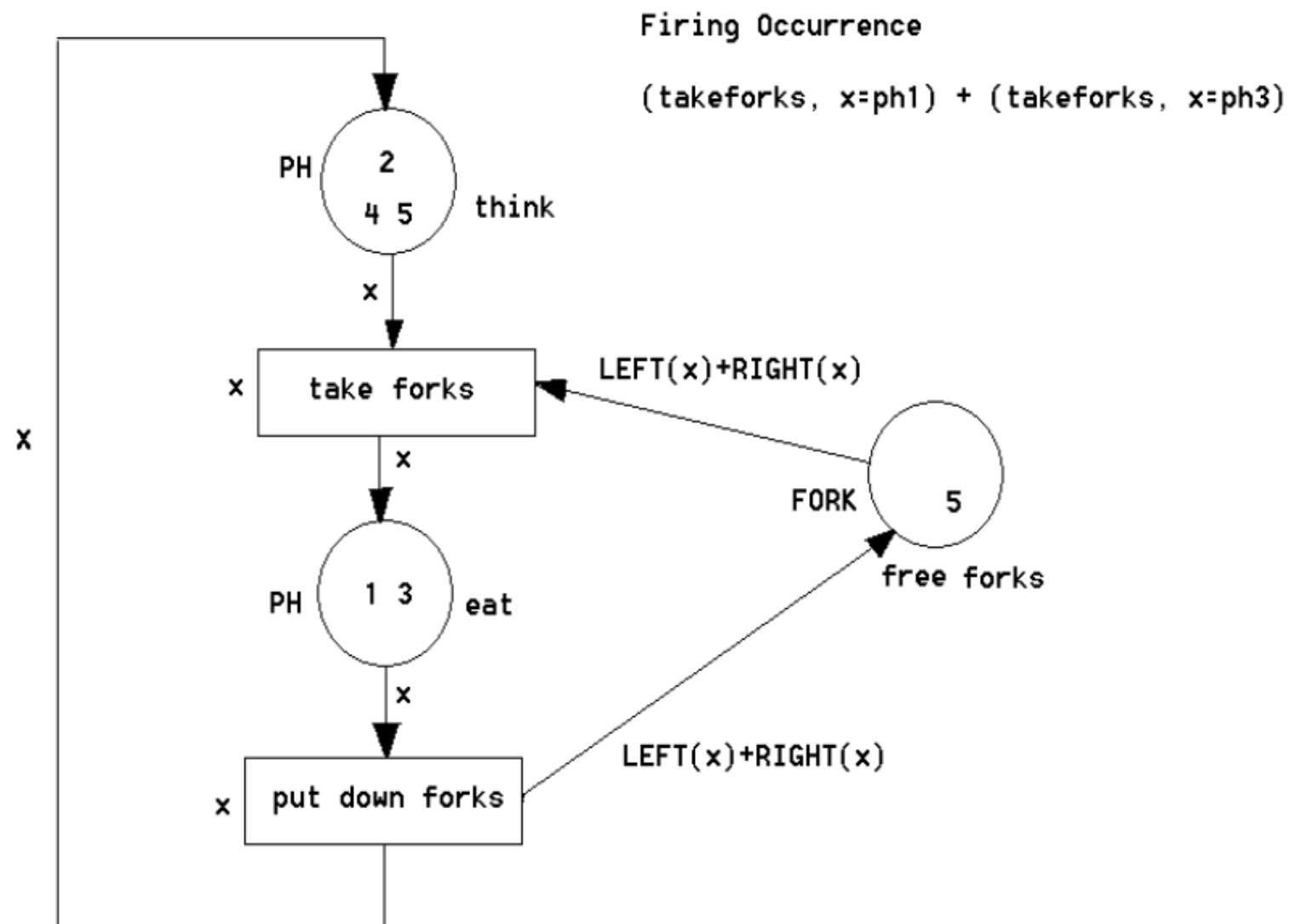
LEFT: PH->FORK RIGHT: PH->FORK

fun LEFT x=case x of ph1=>f2|ph2=>f3|ph3=>f4
ph4=>f5 | ph5=>f1

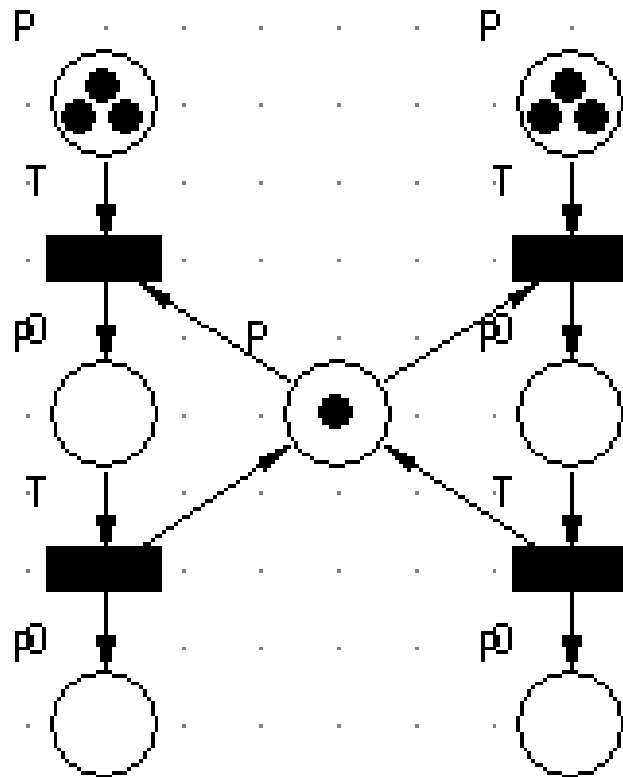
fun RIGHT x=case x of ph1=>f1|ph2=>f2|ph3=>f3
ph4=>f4|ph5=>f5



After Firing:

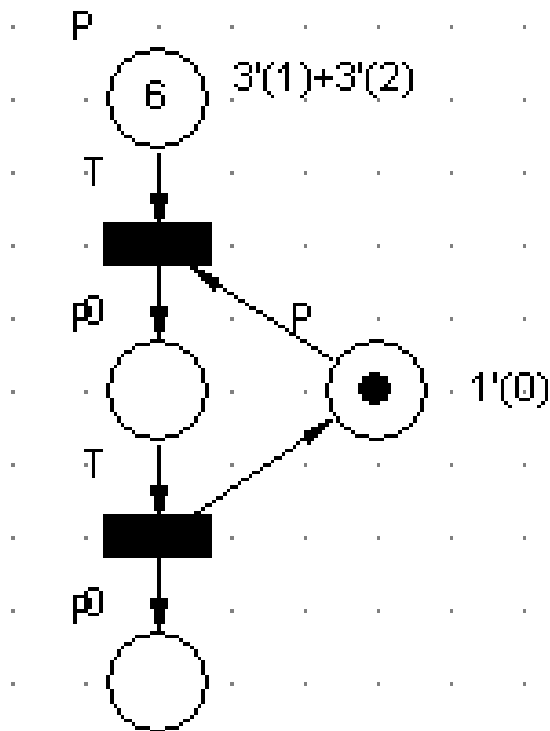


Exemple XdPC: Procés Paral·lel



- Stock inicial de peces A i B
- Recurs compartit per processar-les

Exemple XdPC: Procés paral·lel

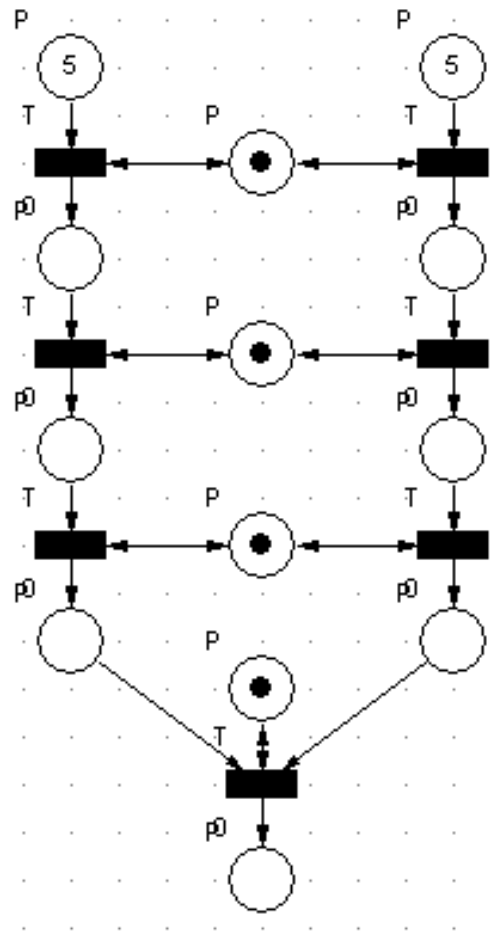


- Color peça = $\{1,2\}$
- Color màquina = $\{0\}$
- Var p : màquina
- Var $p0$: peça

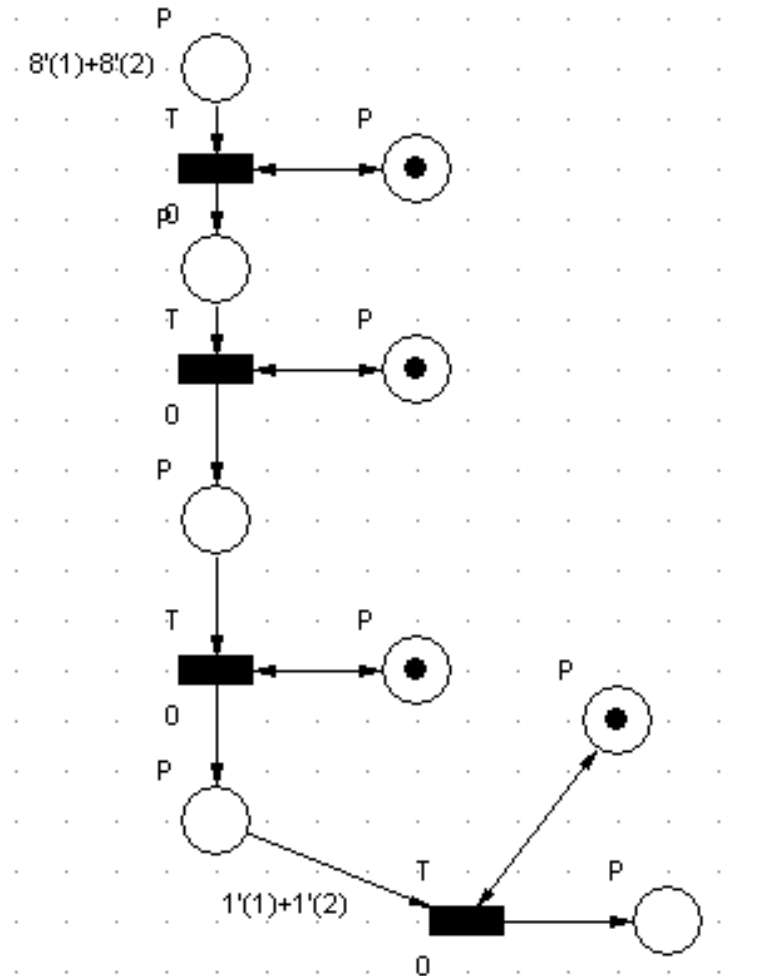
Exemple: Procés productiu 1

- 4 màquines
- 2 tipus de peces, A,B
- S'han de fer les mateixes operacions (una per màquina)
- El temps d'operació depèn del tipus de peça
- L'operació final uneix les peces A i B

Exemple: Procés productiu 1 XdP



Exemple: Procés productiu 1 XdPC



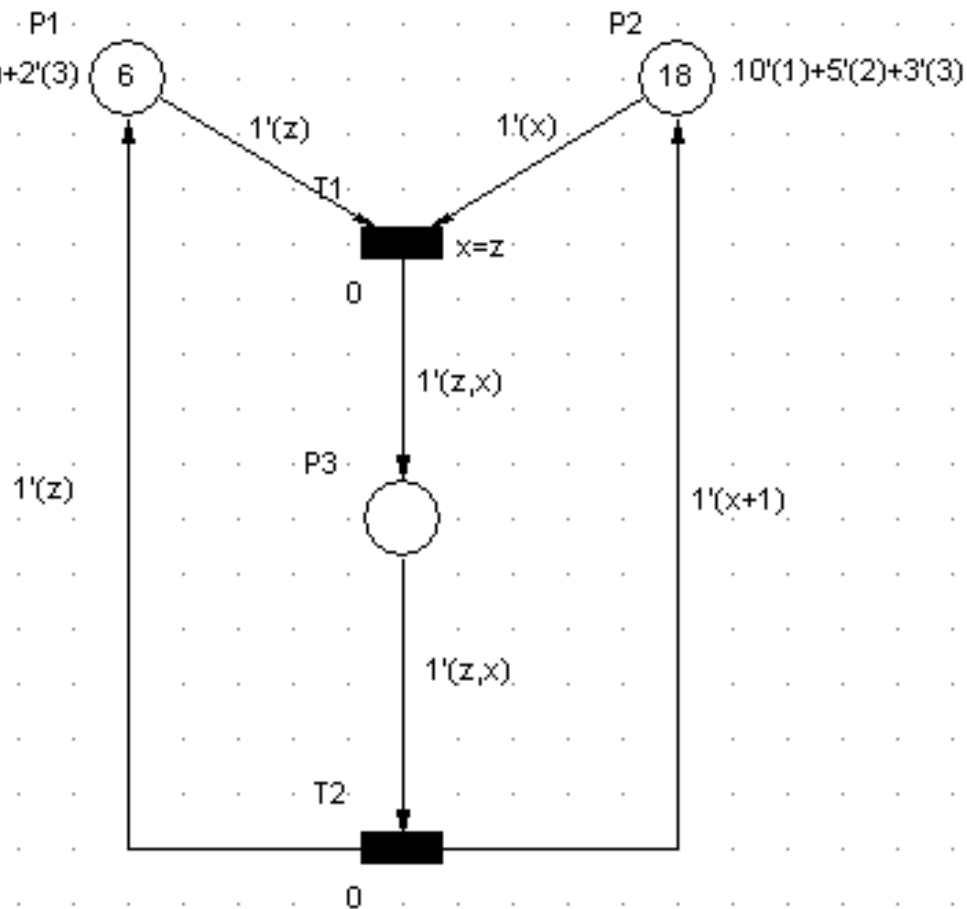
Exemple: Procés productiu 2

- 3 tipus de màquines. Cada conjunt de dues màquines té una cua inicial.
 - ▣ 2 màquines d'erosió
 - ▣ 2 màquines de mecanitzat
 - ▣ 2 màquines d'assemblatge
- Peces que han de fer aquestes operacions en aquest ordre.
- Estat inicial: 10 peces a la cua 1, 5 a la cua 2 i 3 a la cua 3.

Exemple: Procés productiu 2

- x : conjunt de colors de valors 1,2,3,4 que indica si esta en la cua 1, 2, 3 o 4.
- z : tipus de màquina (1, erosió; 2 mecanitzat; 3, assamblat)
- R : producte cartesià $z \times x \times x$

Exemple: Procés productiu 2



P1: màquines lliures

P2: peces pendents de processar

P3: peces processant-se

Color pos_peça = $\{1,2,3,4\}$

Color màquina = $\{1,2,3\}$

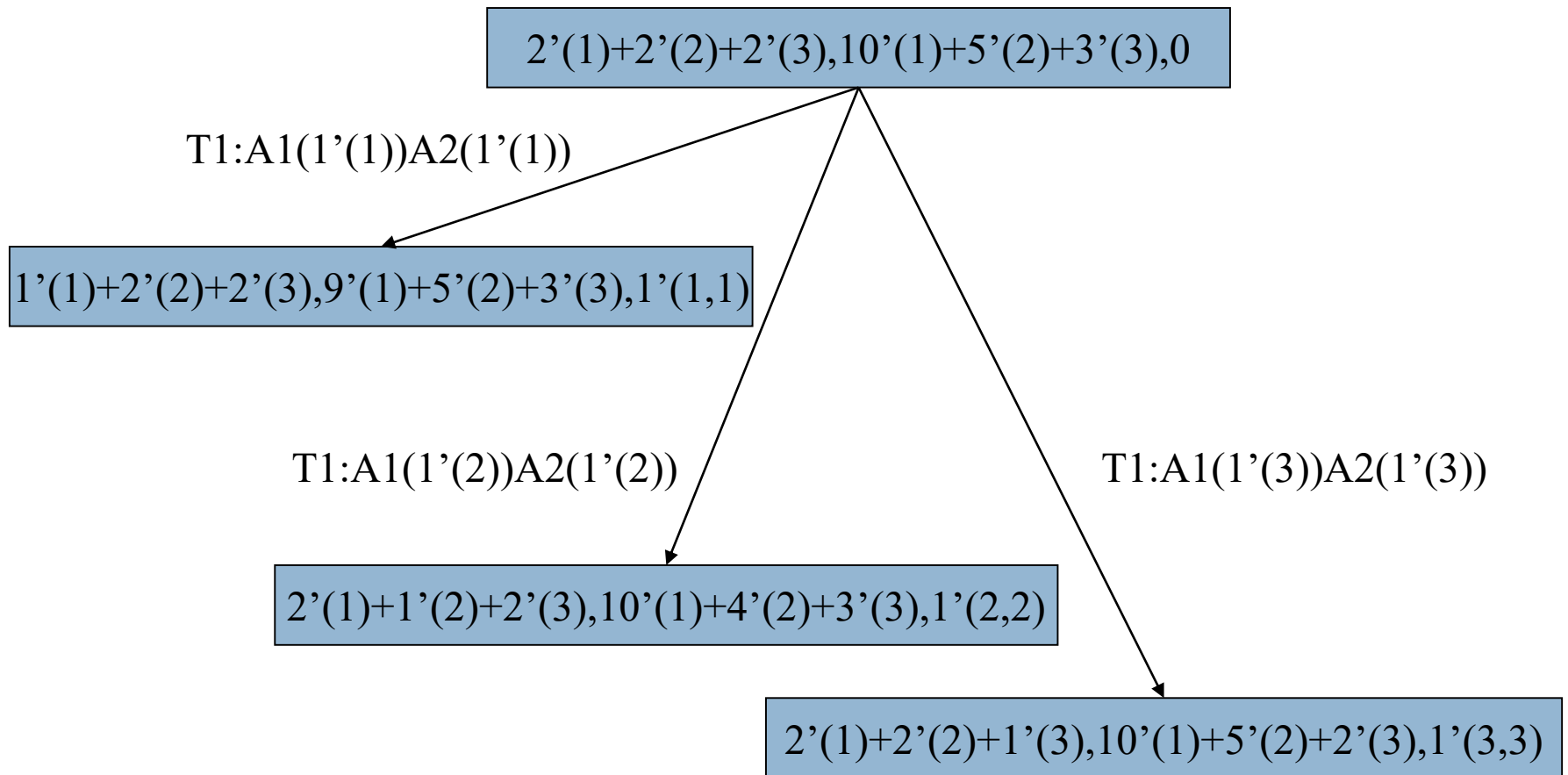
Var z: màquina

Var x: peça

Exemple: Procés productiu 2

- $M0 = [2'(1) + 2'(2) + 2'(3), 10'(1) + 5'(2) + 3'(3), 0]$
- Construcció del arbre d'abast.

Exemple: Procés productiu 2



Xarxes de petri temporals

XdPT

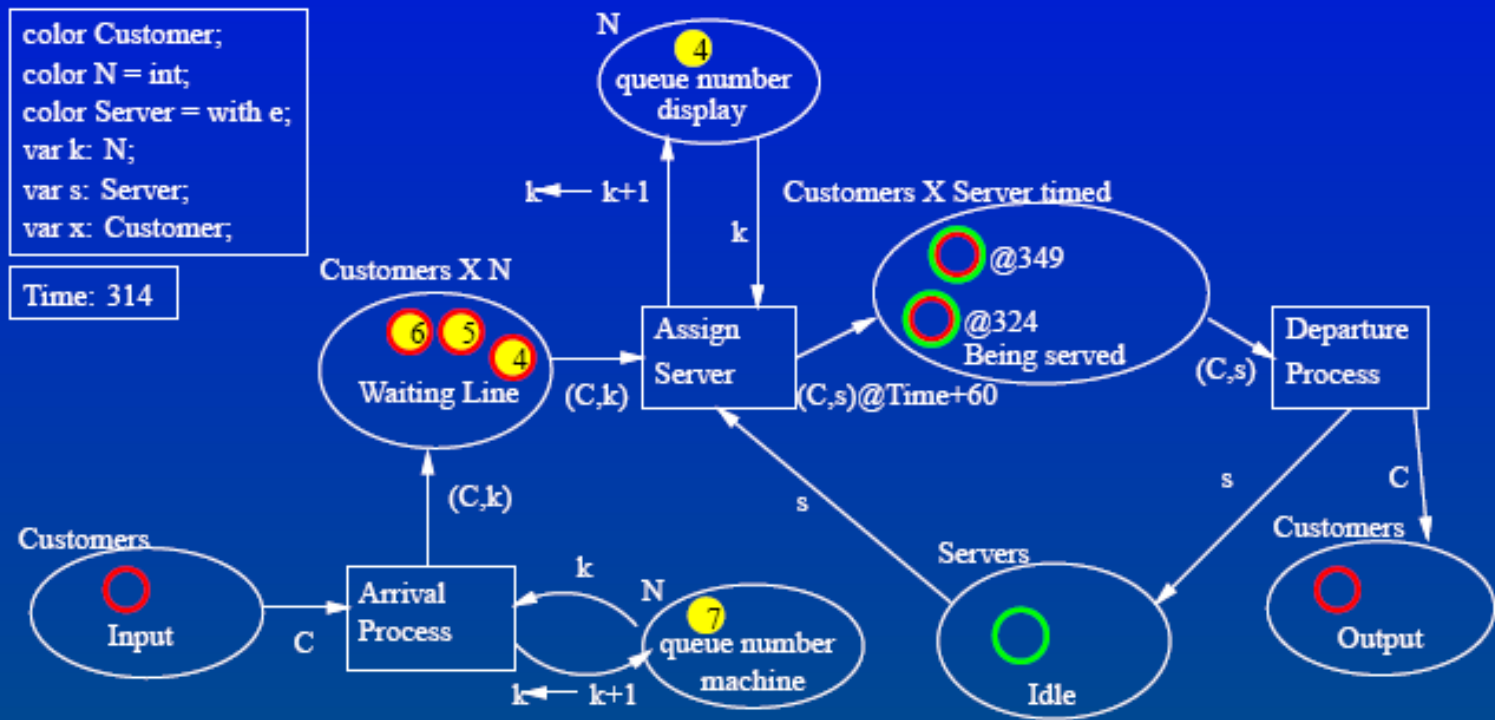
XdPT: Motivació

- En el món real es necessita esperar per un servei, o el servei triga un temps en executar-se.
- El temps ha d'afegir-se a les xarxes de Petri per tal de fer una anàlisi de rendiment.

Xarxes de Petri amb temps: XdPT

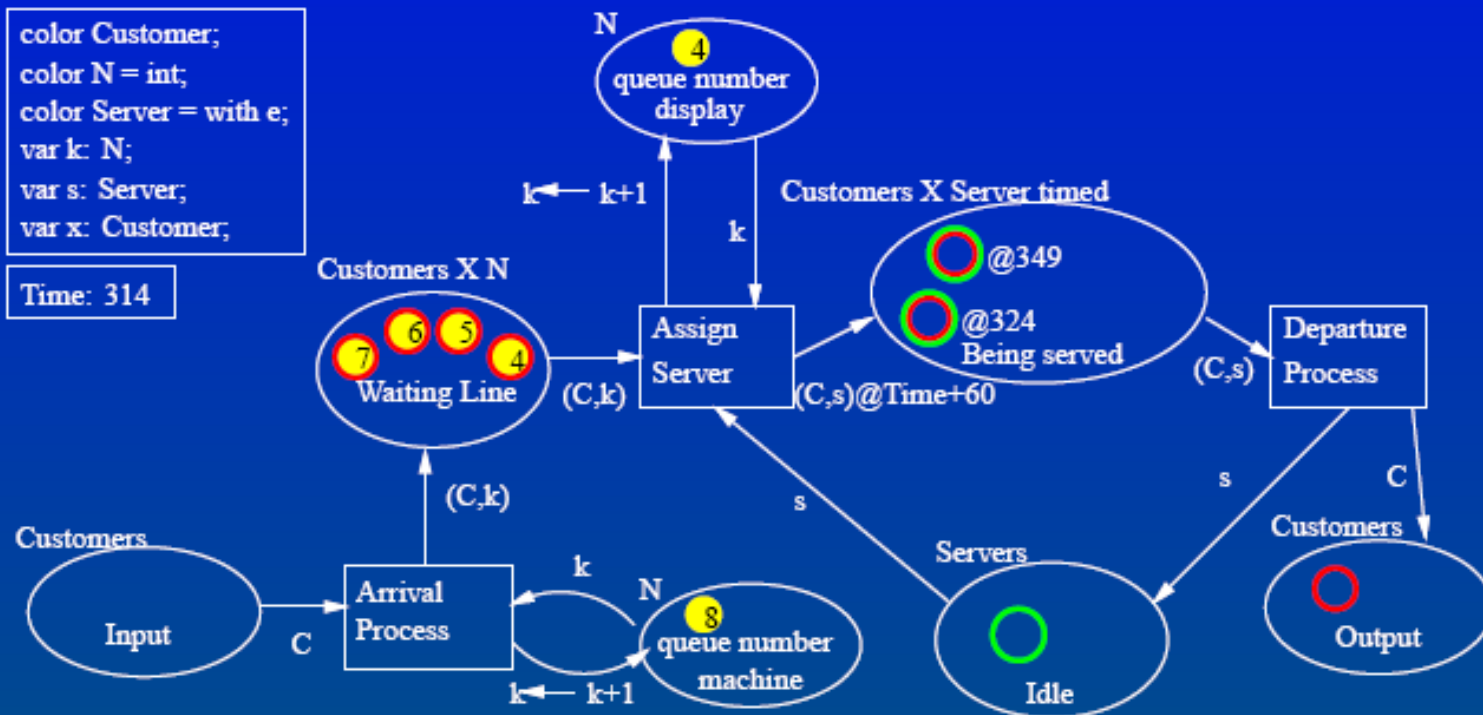
- Les transicions tenen associat un temps.
- Totes les marques poden portar un identificador temporal “time stamp”, usualment el temps de creació
- En una xarxa de petri temporal existeix un rellotge global que representa el temps del model.

A G/D/3 FIFO queue in TCPN



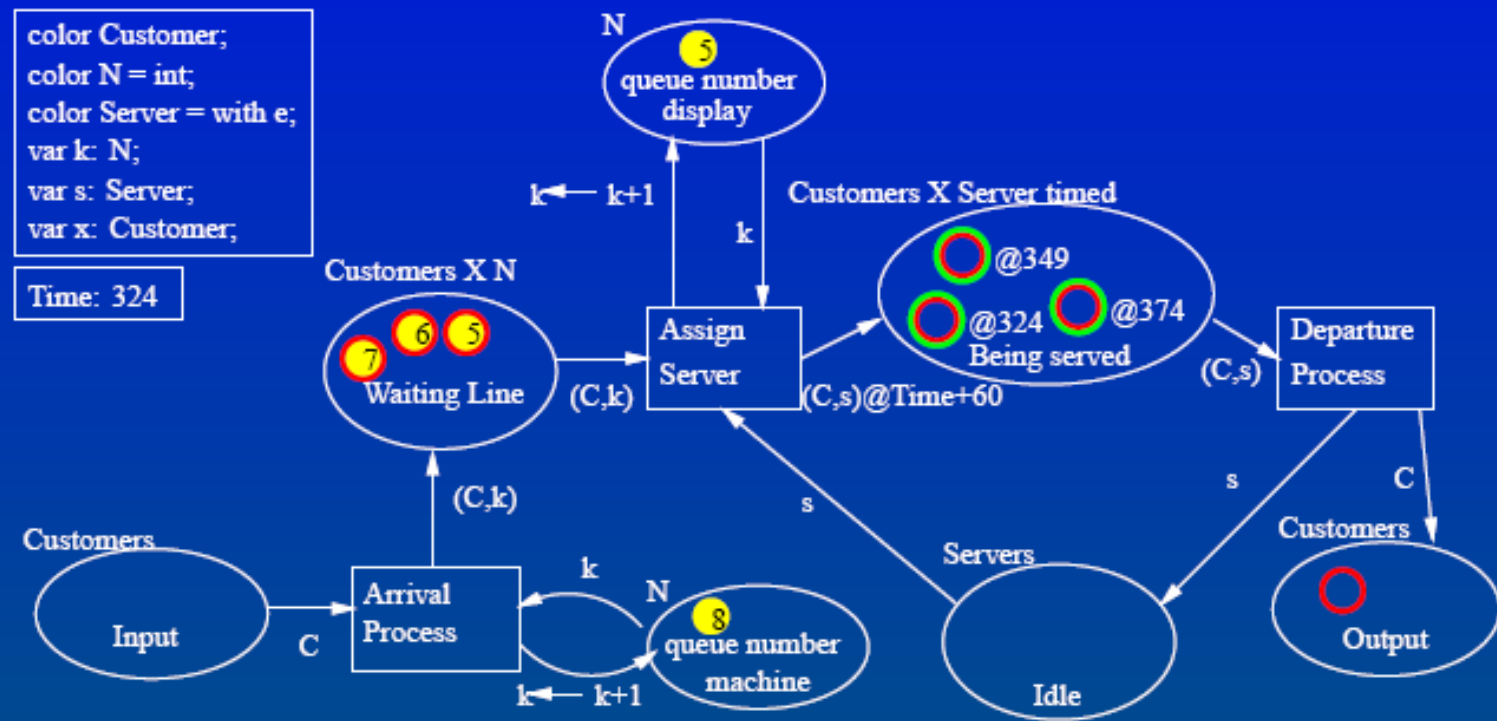
Time is added to the "Being served"-place. All immediate firing is done first.

A G/D/3 FIFO queue in TCPN



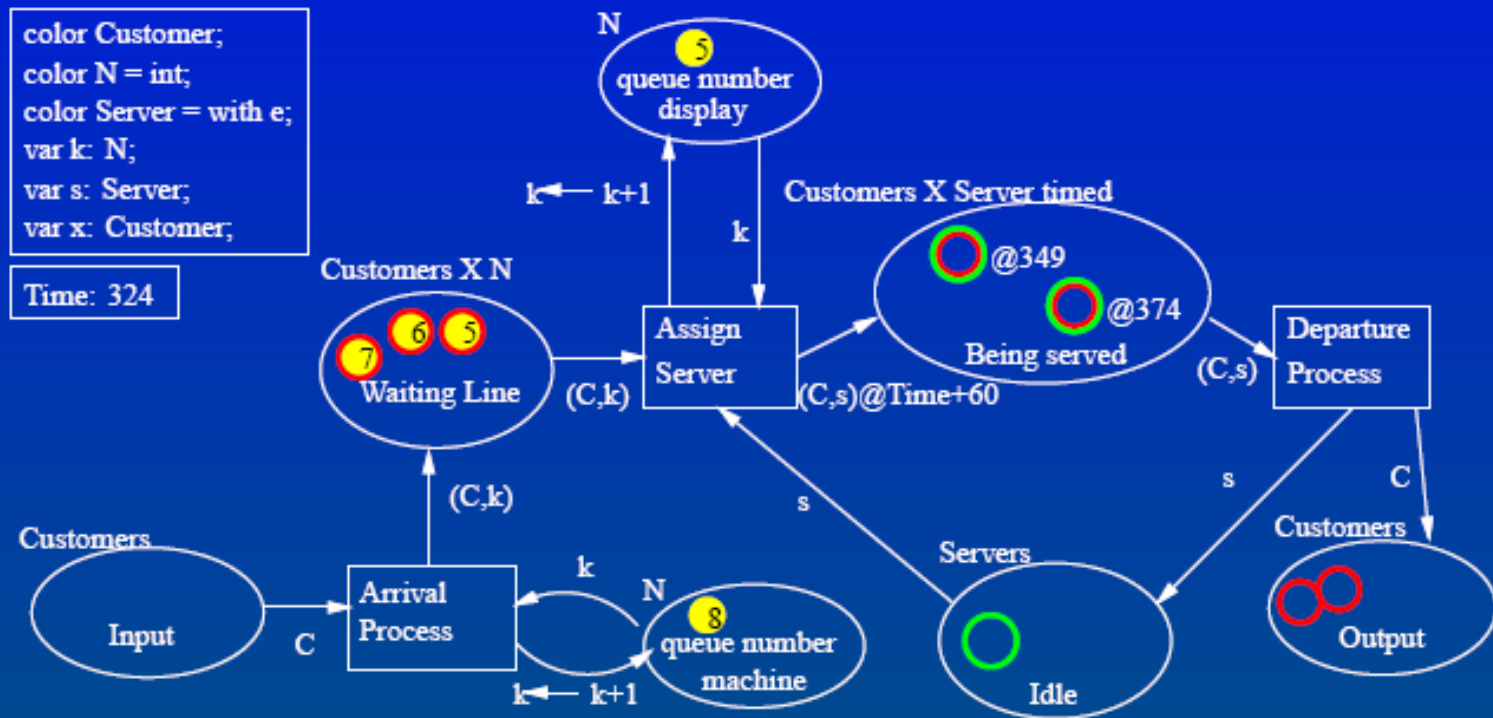
Notice that a server is idle.

A G/D/3 FIFO queue in TCPN

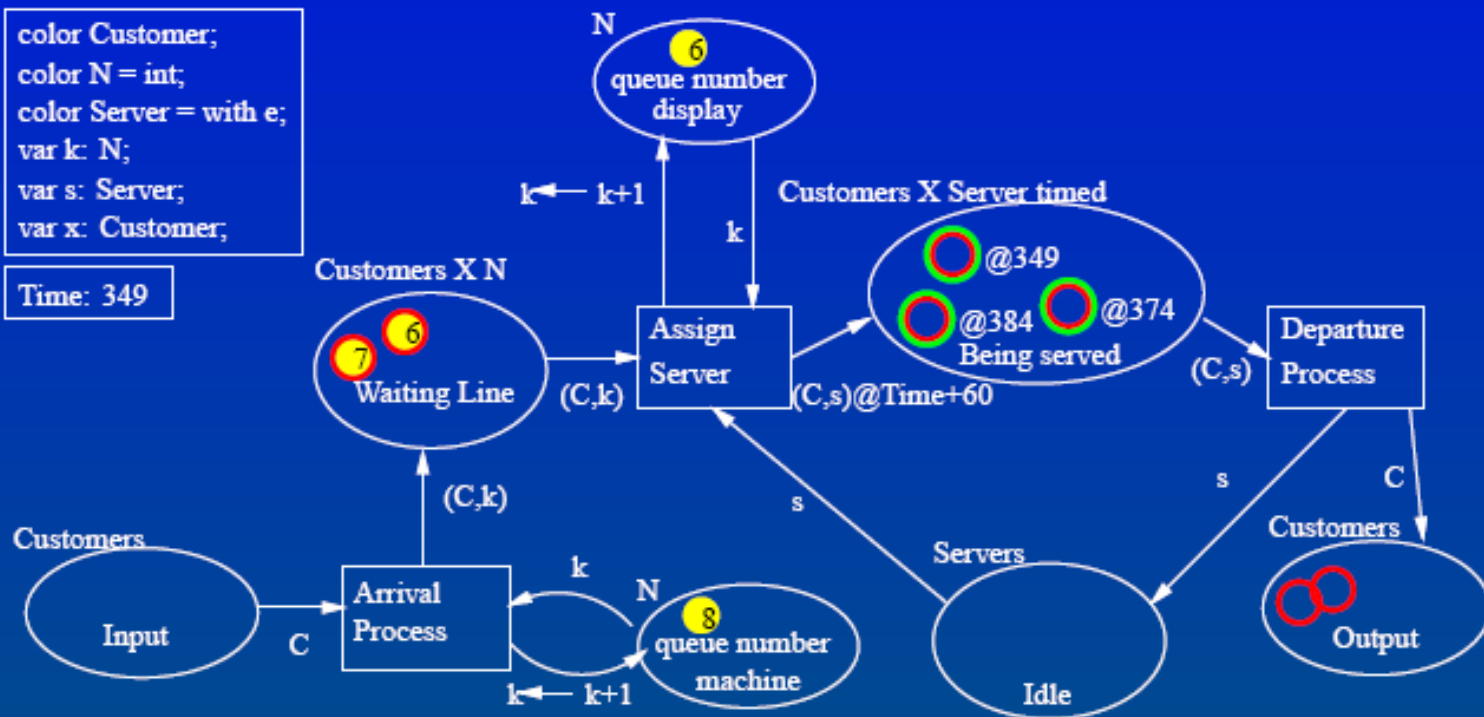


The customer with the timestamp @324 is ready.

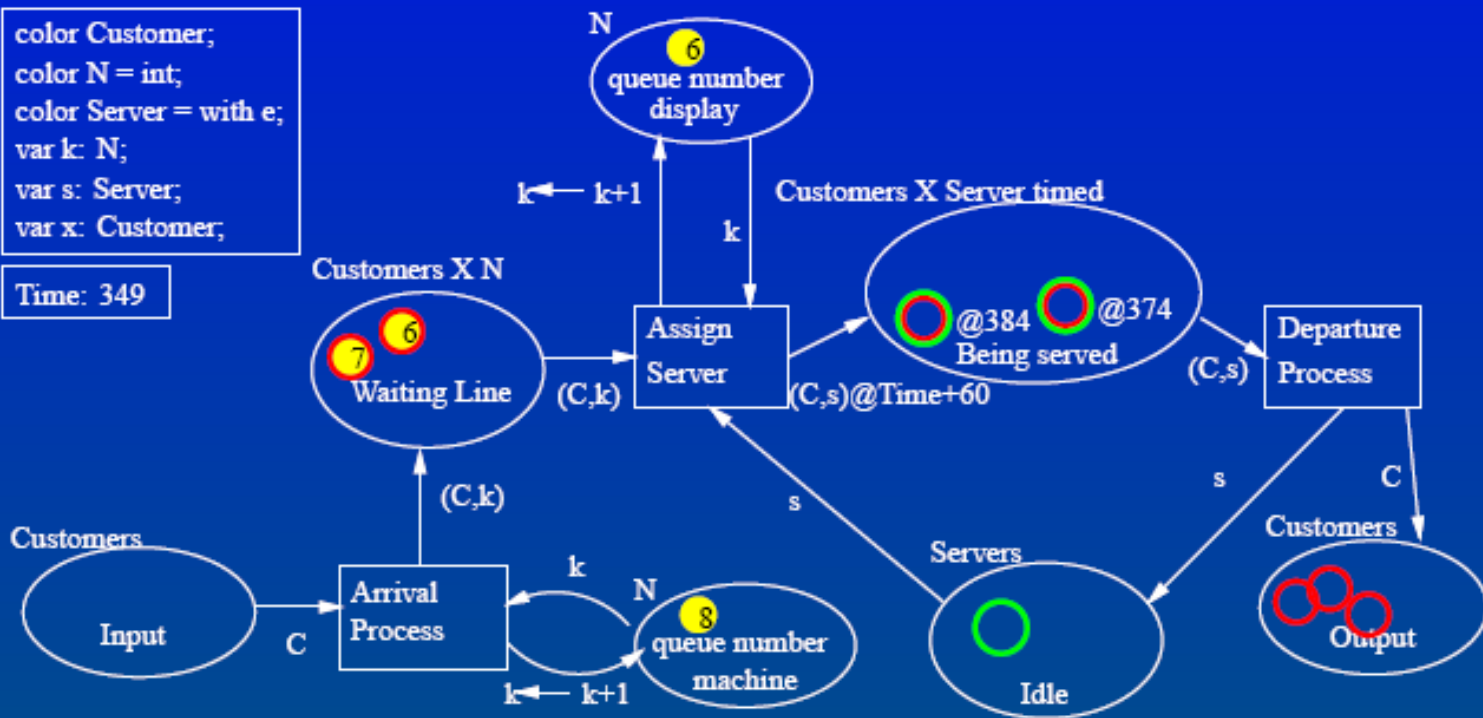
A G/D/3 FIFO queue in TCPN



A G/D/3 FIFO queue in TCPN



A G/D/3 FIFO queue in TCPN

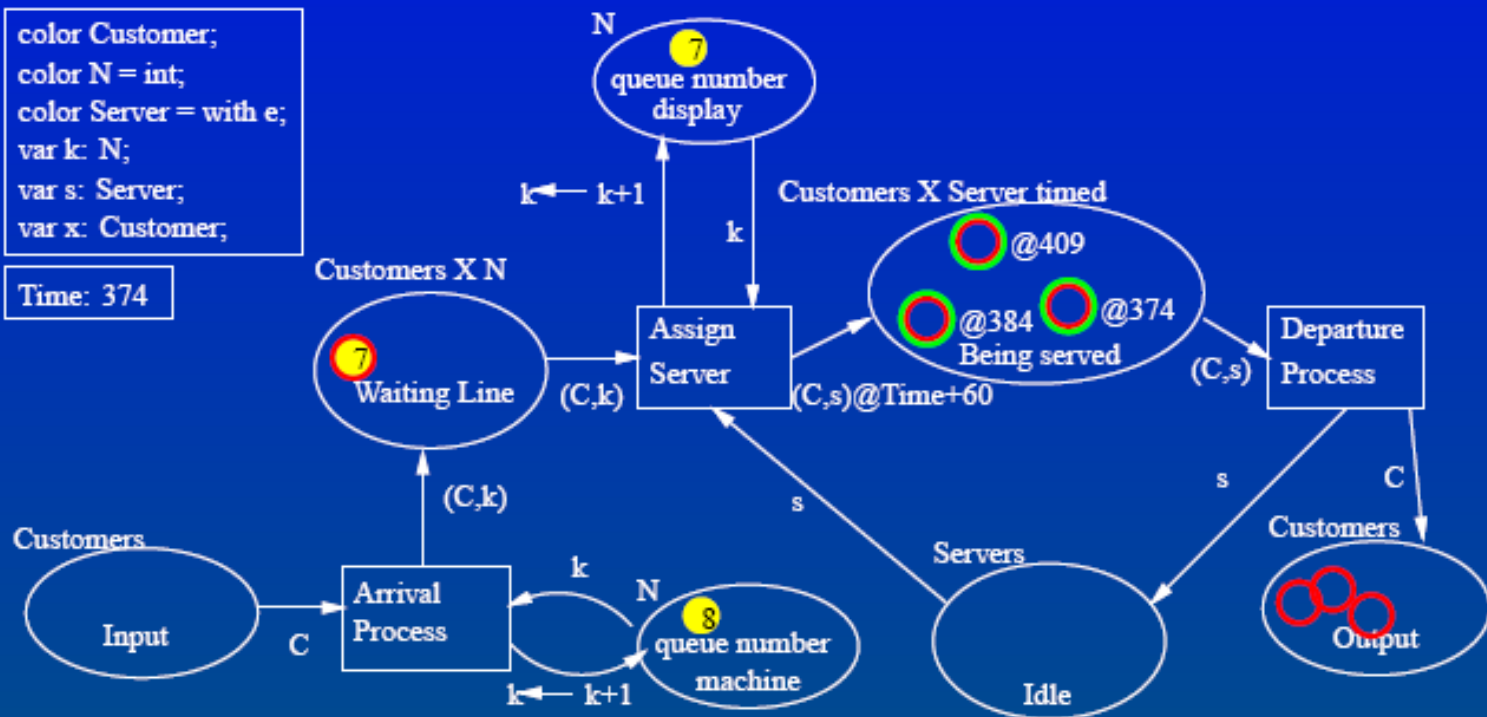


A G/D/3 FIFO queue in TCPN

```

color Customer;
color N = int;
color Server = with e;
var k: N;
var s: Server;
var x: Customer;
    
```

Time: 374



Xarxes de Petri estocàstiques

XdPEst

XdPEst: motivació

- Afegir l'aleatorietat en les especificacions dels models.

XdPEst: Tipologies

- Xarxes de petri estocàstiques.
- Xarxes de petri estocàstiques generalitzades.
- Xarxes de petri deterministes estocàstiques.

XdPEst:

- Tots les demores de les transicions estan distribuïdes exponencialment per variables aleatòries.

XdPEst: Xarxes de petri estocàstiques generalitzades

- Demores en els disparadors temporals estan distribuïdes amb variables aleatòries.
- Transicions immediates amb temps zero amb prioritat sobre les transicions temporals.
- Model senzill d'implementar.

XdPEst: Xarxes de petri deterministes estocàstiques

- Demores en els disparadors temporals estan distribuïdes amb variables aleatòries.
- Les demores de les transicions deterministes son variables deterministes.
- Transicions immediates amb temps zero amb prioritat sobre les transicions temporals.

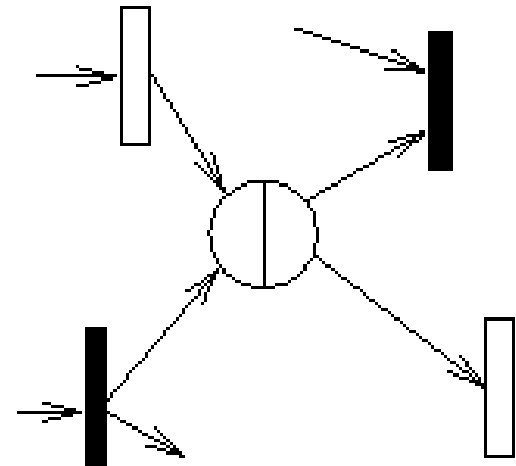
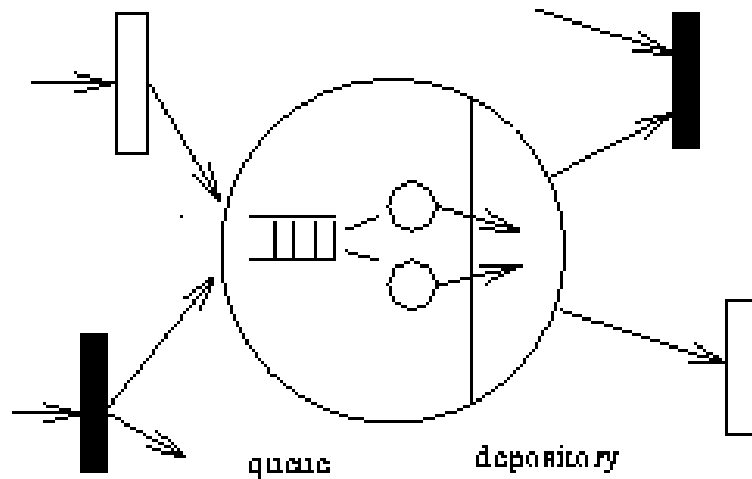
XdPEst: Anàlisi

- Fracció de temps esperada gastada en un subconjunt de marques.
- Nombre esperat de marques en un determinat lloc.
- Nombre esperat de transicions en un determinat interval temporal
- Freqüència de transicions activades.

Xarxes de petri amb cues

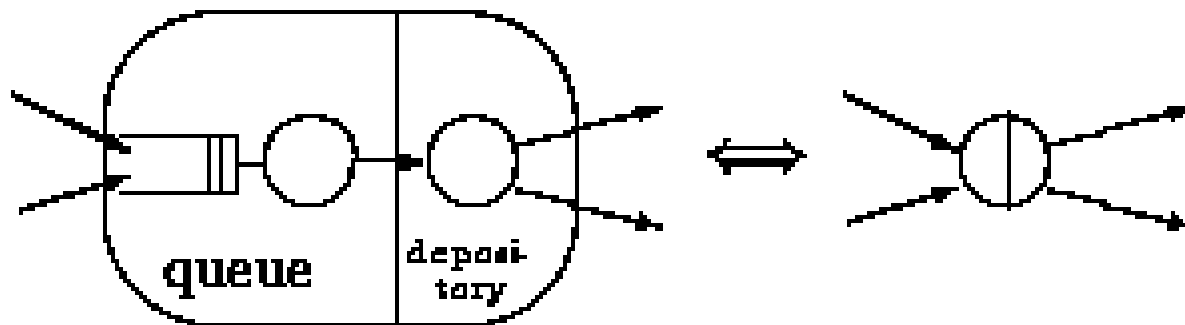
XdPQ

Xarxes de Petri amb cues: XdPQ



XdPQ

- Adaptació de les CPM
- Aspectes temporals i funcionals per els llocs



Xarxes de petri variants

XdPV

Xarxes de Petri Variants

- Xarxes de Petri Temporitzades:
 - ▣ Temps associats amb les transicions i les posicions.
- Xarxes de Petri d' alt nivell:
 - ▣ Les marques son tipus de dades estructurats. (objectes).
- Xarxes de Petri Híbrides & Continues:
 - ▣ On les marques son nombres reals en comptes d'enters.
- Barreja de sistemes continus/sistemes discrets.

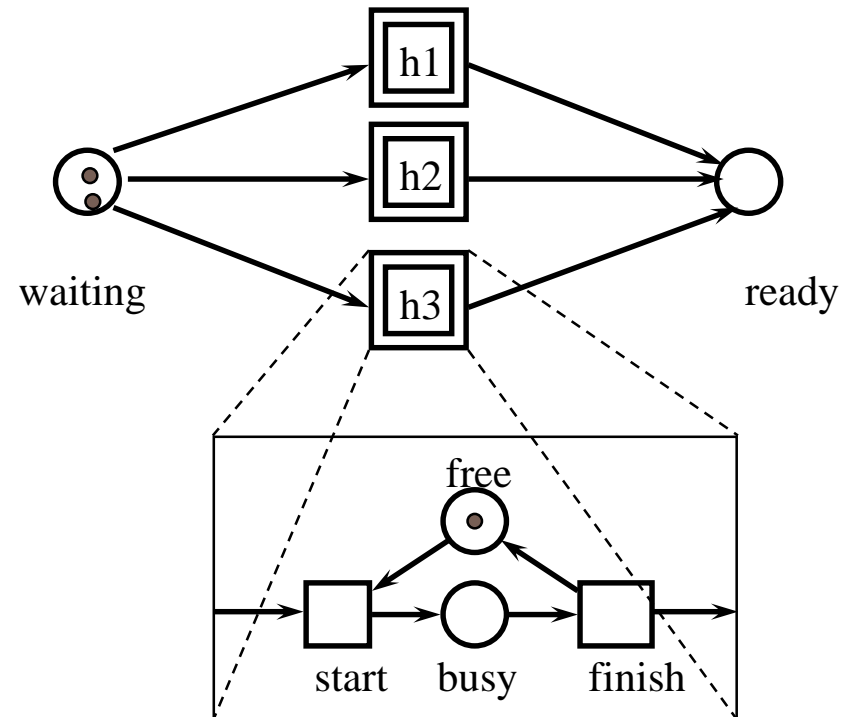


Jerarquia a les xarxes de Petri

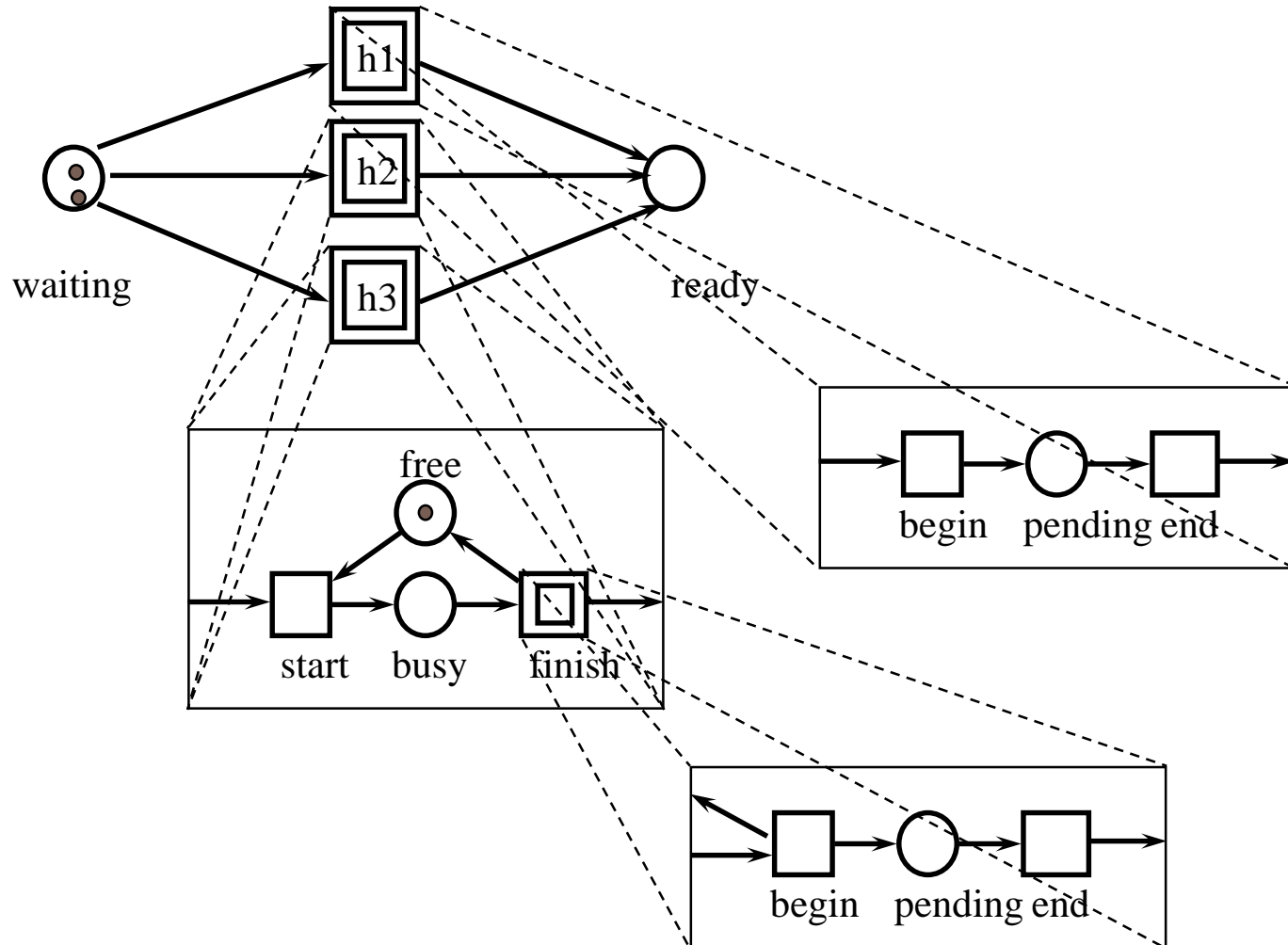
Tractament de la modularitat

The extension with hierarchy

- A mechanism to structure complex Petri nets.
- A **subnet** is a net composed out of places, transitions and subnets.



Hierarchy example



Exemple

- Dos processos paral·lels, A i B.
- Comparteixen un robot que permet moure les peces a una cinta de sortida compartida.
 - ▣ Un dels dos processos paral·lels requereix de una peça que es genera cada X unitats de temps. El robot no mourà la peça en aquest procés a no ser que també tingui aquesta peça.
- Dues màquines idèntiques processen les peces de la cinta.

Webs

- <http://www.informatik.uni-hamburg.de/TGI/PetriNets/>