Automates et réseaux de Petri pour les jeux vidéo

S. Natkin

Objectif

Un modèle pour spécifier des relations d’évolution dans non déterministe dans un jeu:

A un niveau macroscopique (scénario on linéaire)

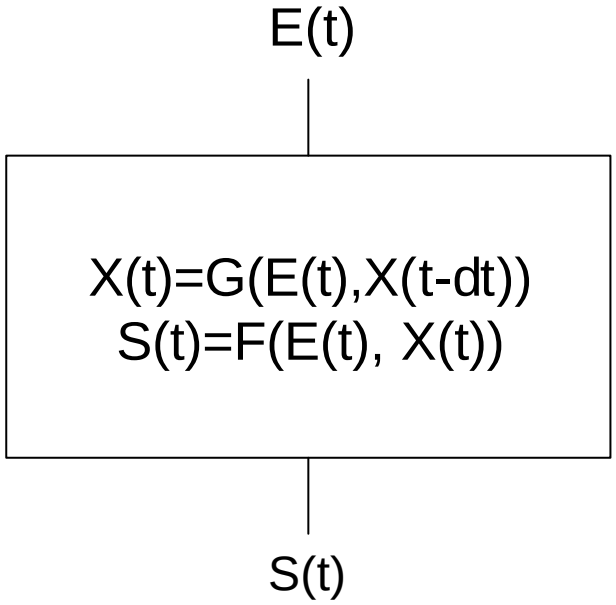
Au niveau du comportement et des interactions entre objets

Plan

- systèmes dynamiques
- Automates à états finis communiquants
- Réseaux de Petri

Systèmes dynamiques

Modèle Mathématique



Comportement d'un moniteur (synchrone)

Répéter

T:=T+DT

Lire les entrées (clavier, souris, joypad…)
Faire jouer l'ordinateur
} E(t)

Calculer la position de tous les objets
Déterminer l'état de tous les objets généraux
Dater tous les évènements actifs les ranger
dans la liste de simulation
et éliminer les évènements inactifs
Traiter les évènements datés T
} X(t)= G(E(t),X(t-dt))

Synthétiser image et son
Attendre la date T+DT
} S(t)=F(E(t), X(t))

Jusqu'à la fin du jeu

Etat d’un système dynamique

- X(t) Un ensemble minimal de données à l’instant t nécessaires pour calculer en fonction de E(t)
  - Toutes les variables du systèmes à t+dt
  - Les sorties à S(t+dt)

  Dans un jeu il faut clairement construire et identifier l’état: c’est ce qui doit être sauvegardé à un point de reprise

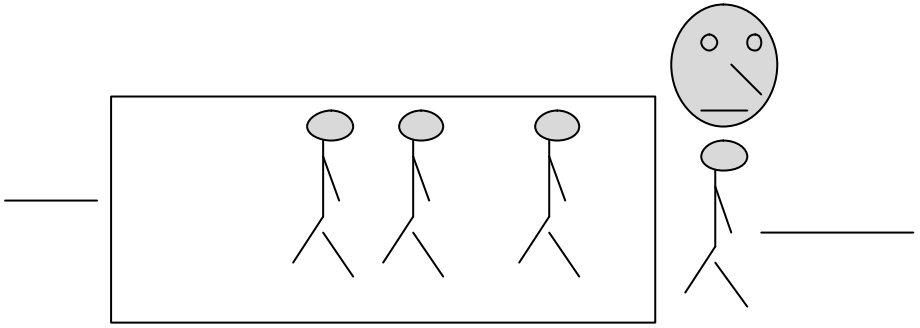
  Si le volume des sauvegarde est limité, il faut choisir les points de sauvegardes qui minimise l’état courant

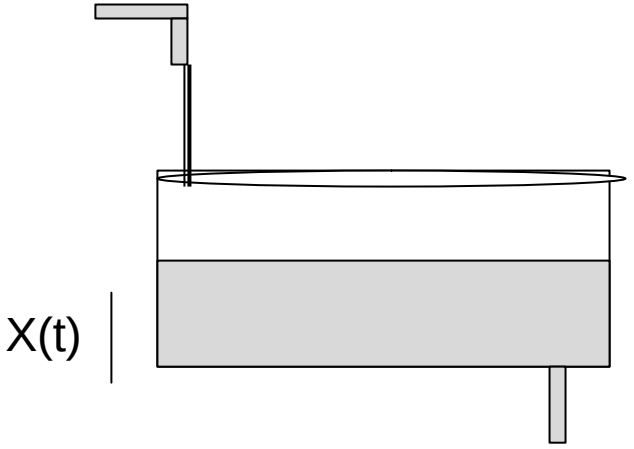
  Par exemple pas de sauvegarde au milieu d’un combat qui nécessiterait de conserver la position de nombreux combattants ou des variables graphiques

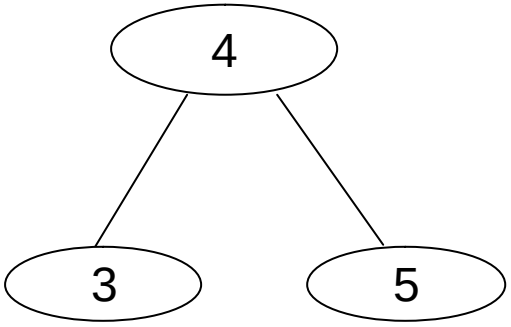
Classes de systèmes

- A espace d'état discret : X(t) est un vecteur de variables à valeur dans un espace discret (dénombrable)
- A espace d'état continu: une au moins des variables de X(t) est continue
- A temps discret: les variables d'états ne se font qu'a des instants dénombrables
- A temps continu: au moins une des variables d'état évolue continuement dans le temps
- Déterministe: a un couple X(t), E(t) correspond un seul état suivant X(t+dt)
- Non déterministe: a un couple X(t), E(t) peut correspondre plusieurs suite possible X(t+dt)

Exemples







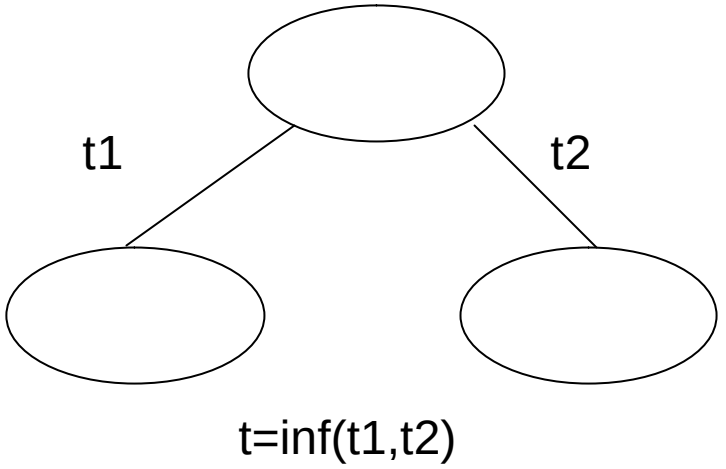
Temps d’attente=X(t)*S

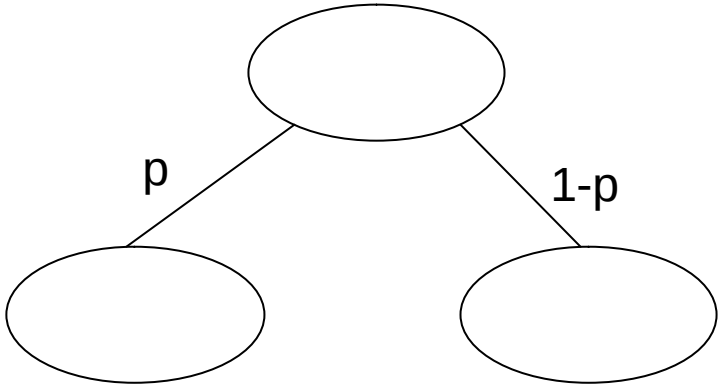
X’(t+dt)=E(t)-S(t)

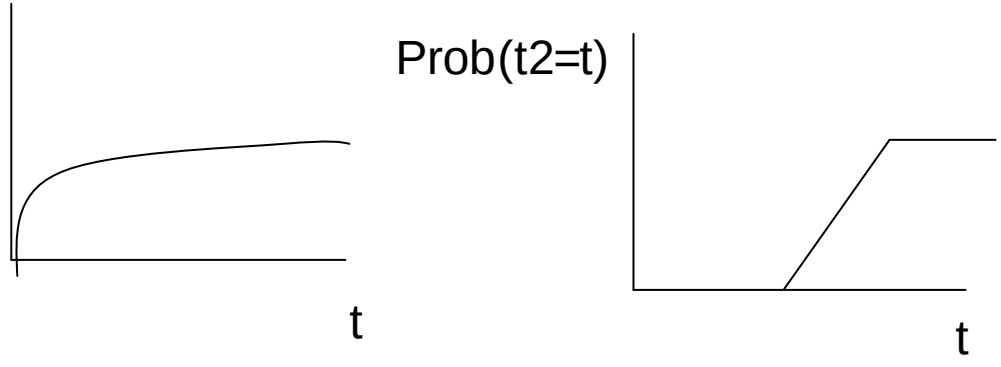
Lever le non déterminisme

- Par le temps: le modèle concurrentiel.
- Par les probabilités
- Par un modèle de temps probabiliste

p= Prob(t1=t2)







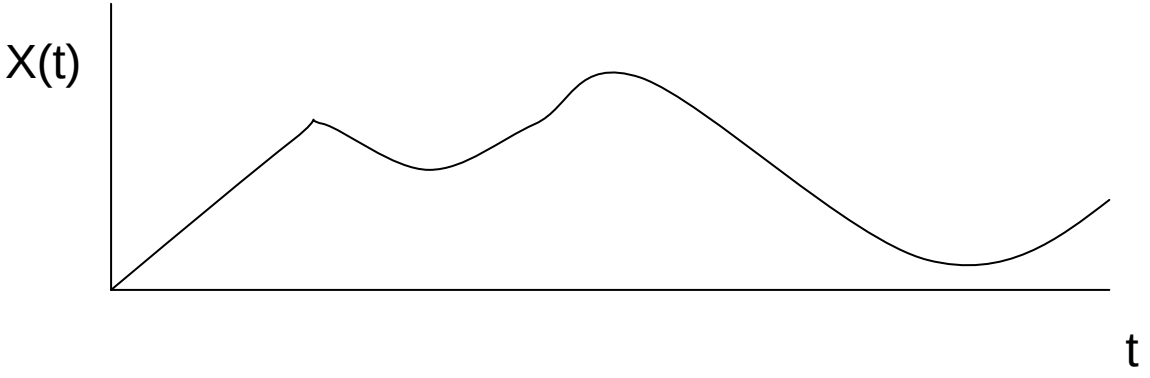
Trajectoire

- Une trajectoire ω est une suite possible d'état dans un intervalle de temps:

ω=(Xo(ω),0), (X1(ω),0),t1),… (Xn(ω),0),tn),…)

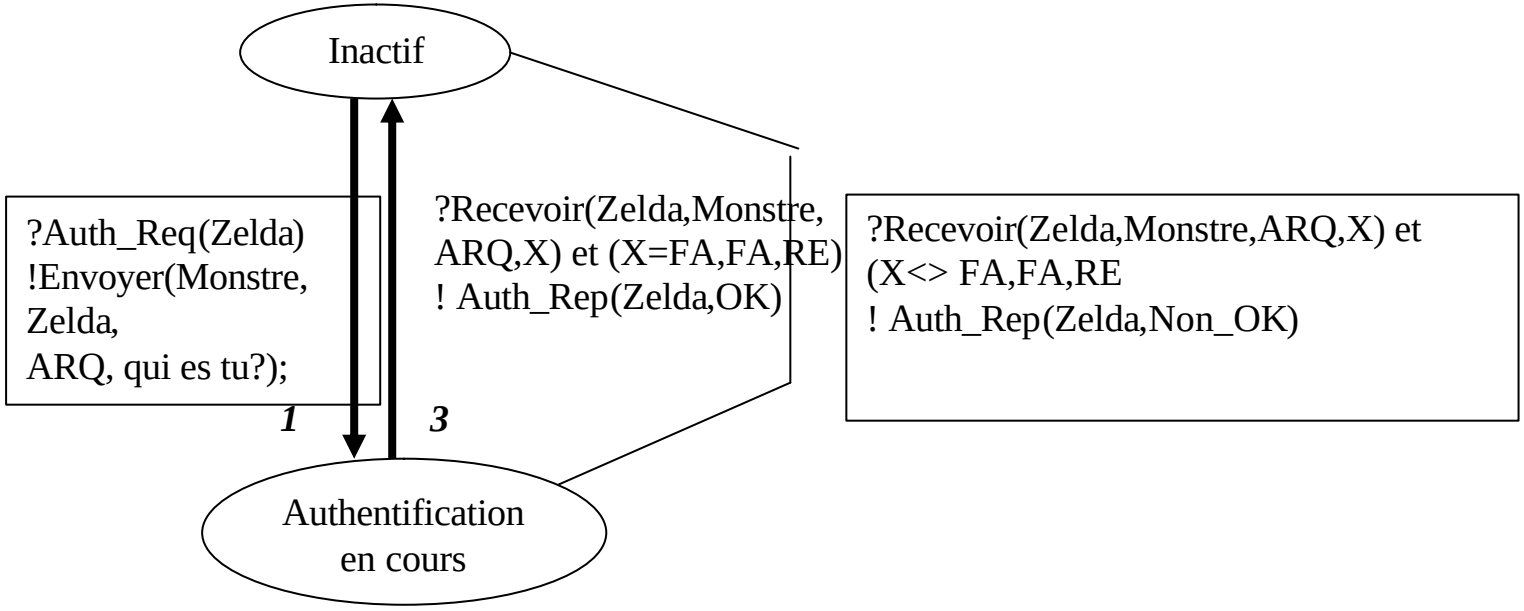
Exemples

(0,0),(1,1.5),(2,2.0),(1,2.8),…

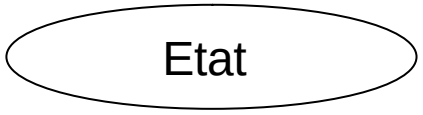


Automates a état finis

Automate du Monstre

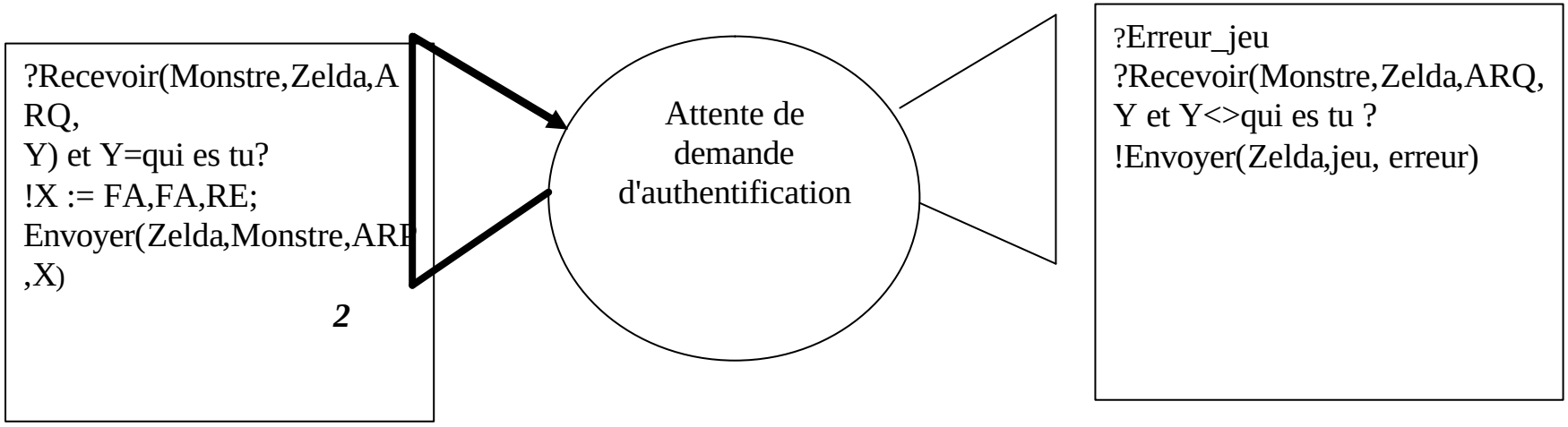


? Langage d'entrée (pré-condition)
! Langage de sortie (post-condition)

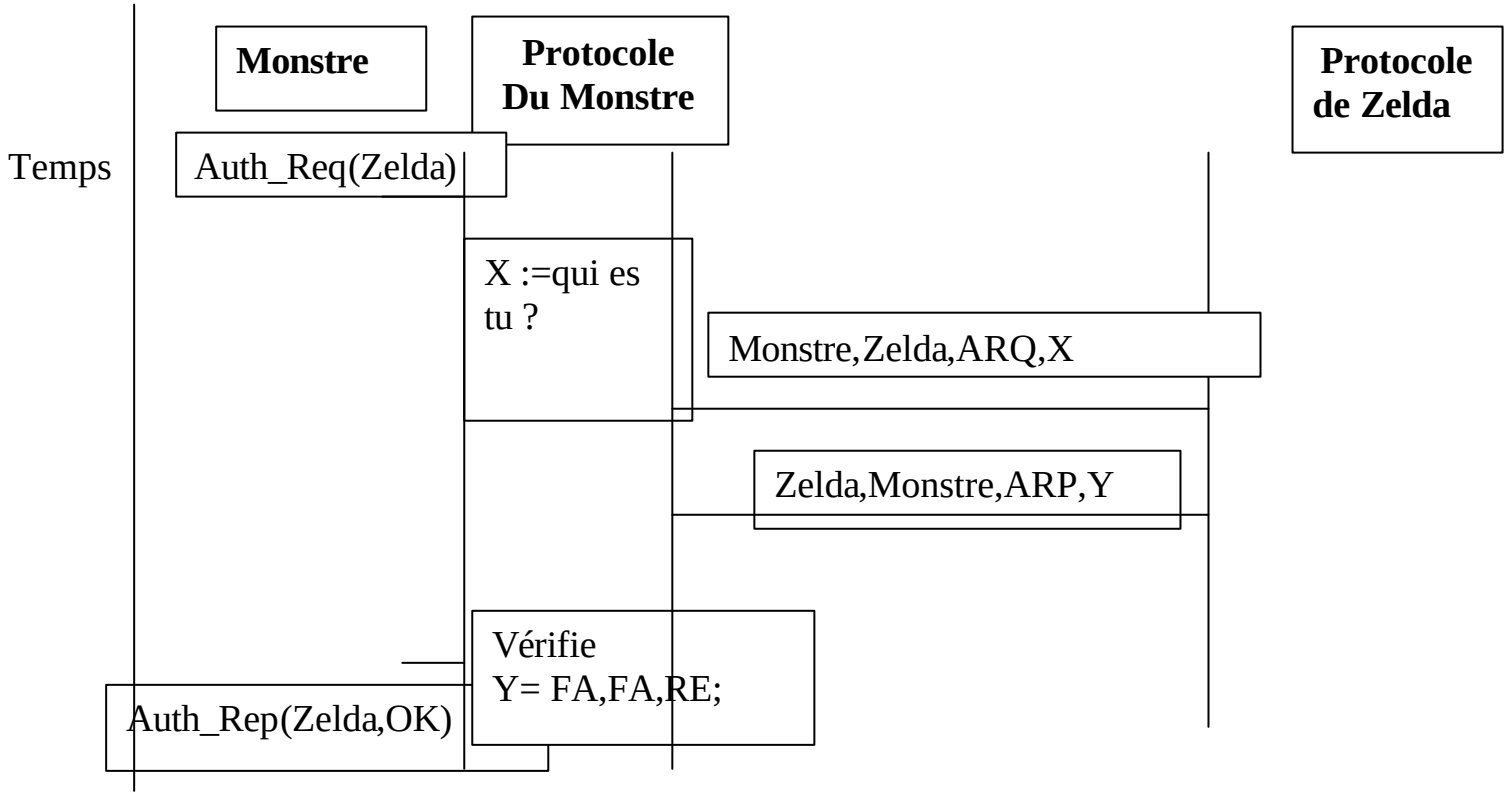


Changement d'état ou transition

Automate de Zelda



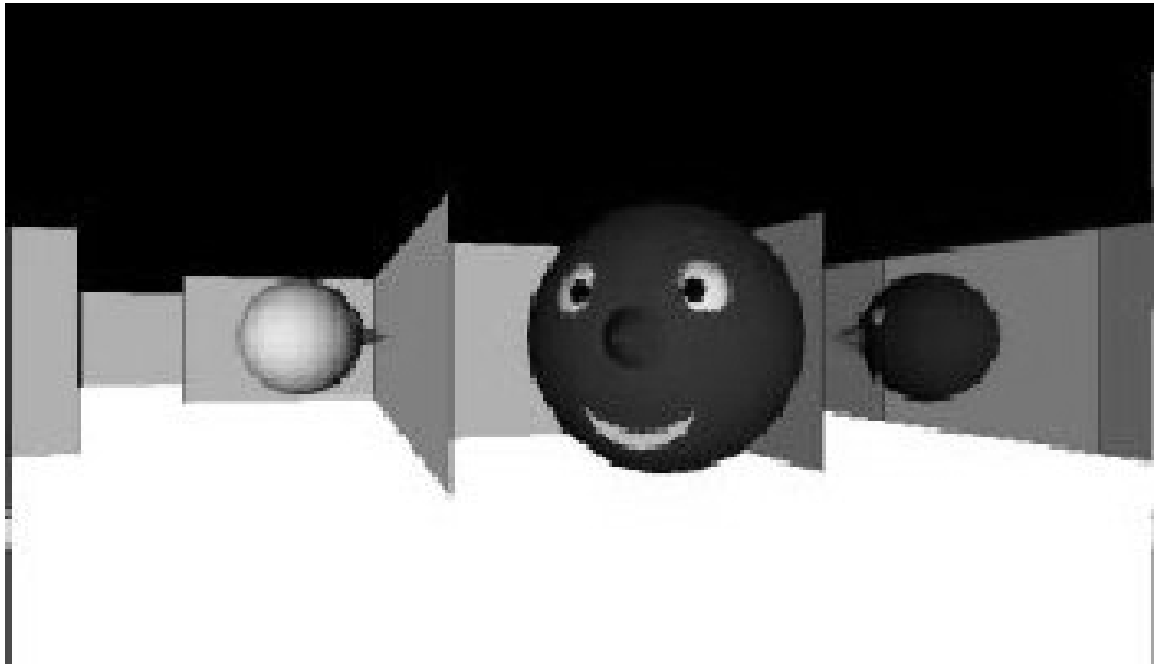
Un diagramme de séquence de messages (trajectoire)



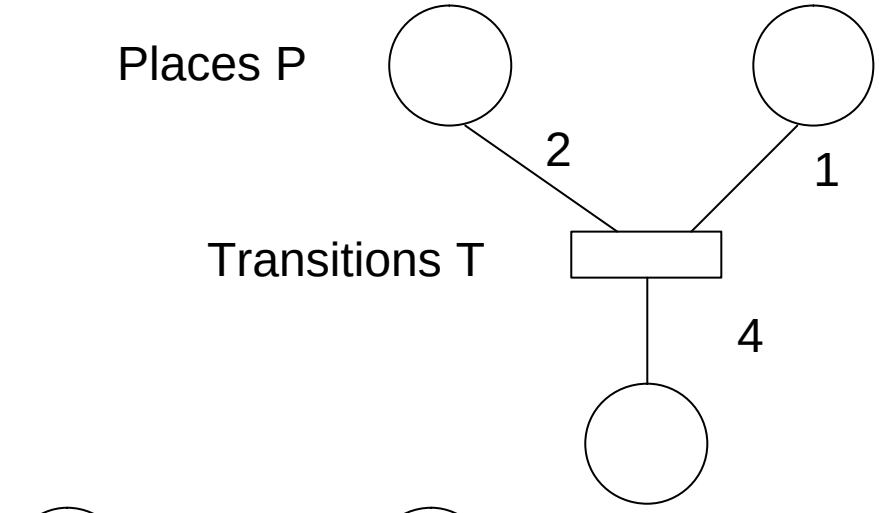
Espace d’état

- Comprend au minimum le produit cartésien de tous les états (automate produit)
- Comprend en outre la valeur de toutes les variables qui apparaissent dans le langage d’entrée et qui ne sont pas des entrées pures
- Comprend toutes les variables qui ne sont pas des entrées pures et qui apparaissent dans le langage de sortie à droite d’une affectation dont la partie gauche est une variable d’état.

Réseaux de Petri

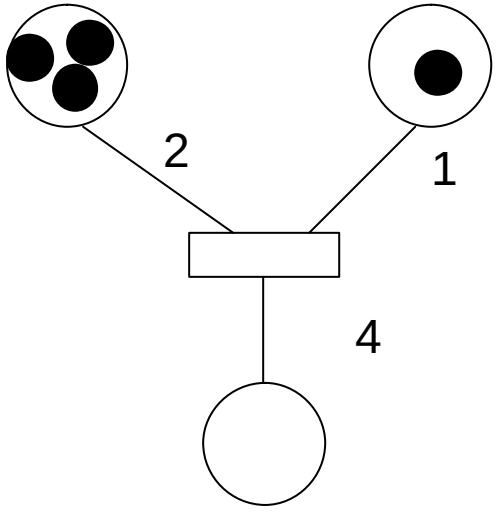


Une version mathématique du jeu de l'oie



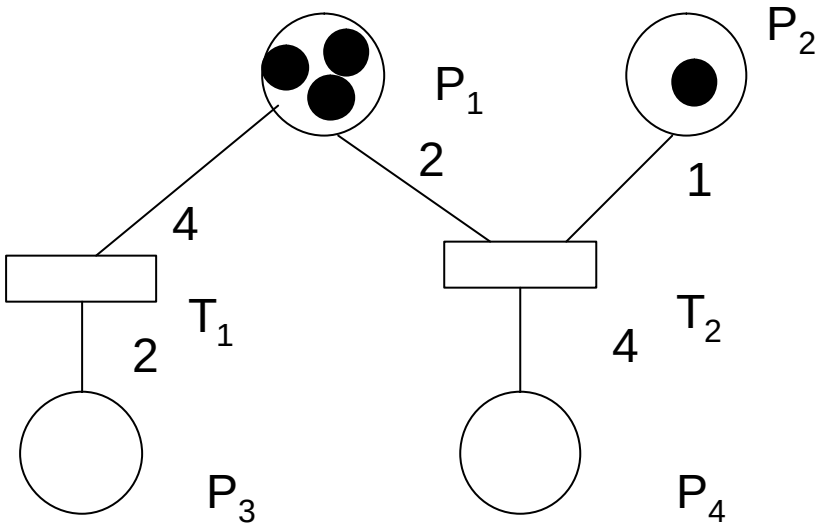
Pré-condition C^-:(PxT) ? N

Post-condition C^+:(TxP) ? N



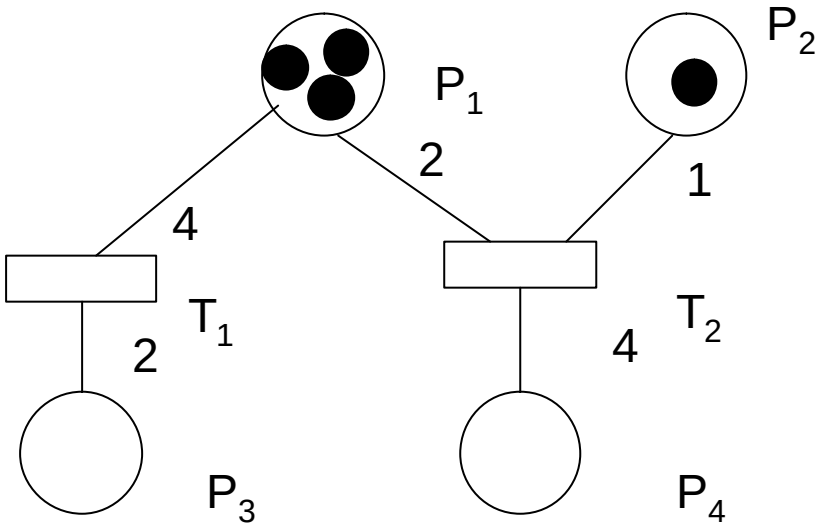
Marquage M ? $N^{card(M)}$

Condition de franchissement



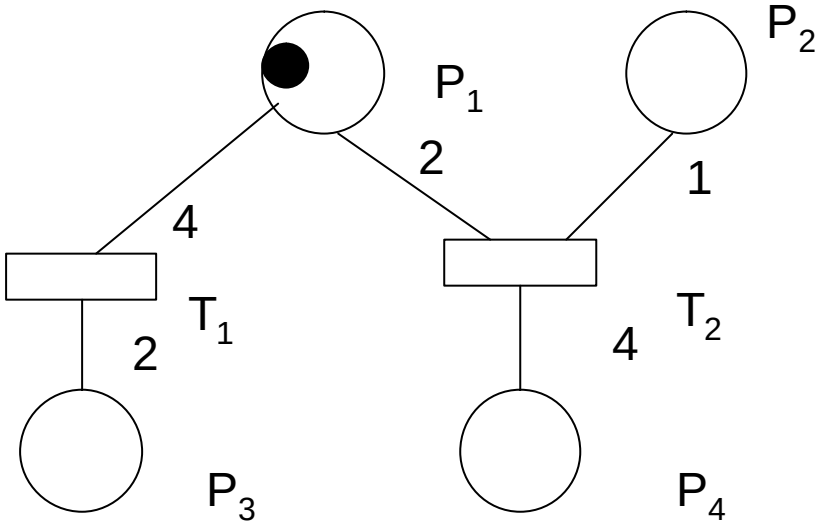
T_j est franchissable si pour toute place P_i $M(i) = C^-(i,j)$

Evolution du Marquage



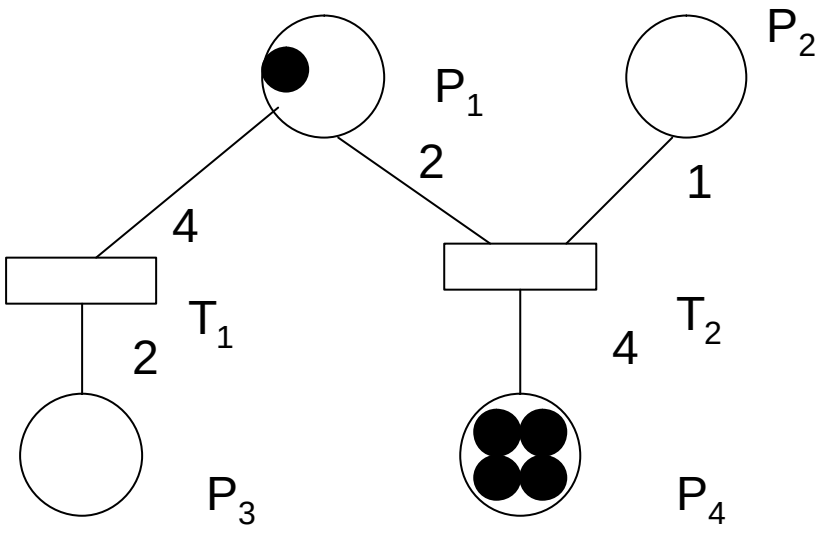
$M' = M - C^{-}(.,j) +$

Evolution du Marquage



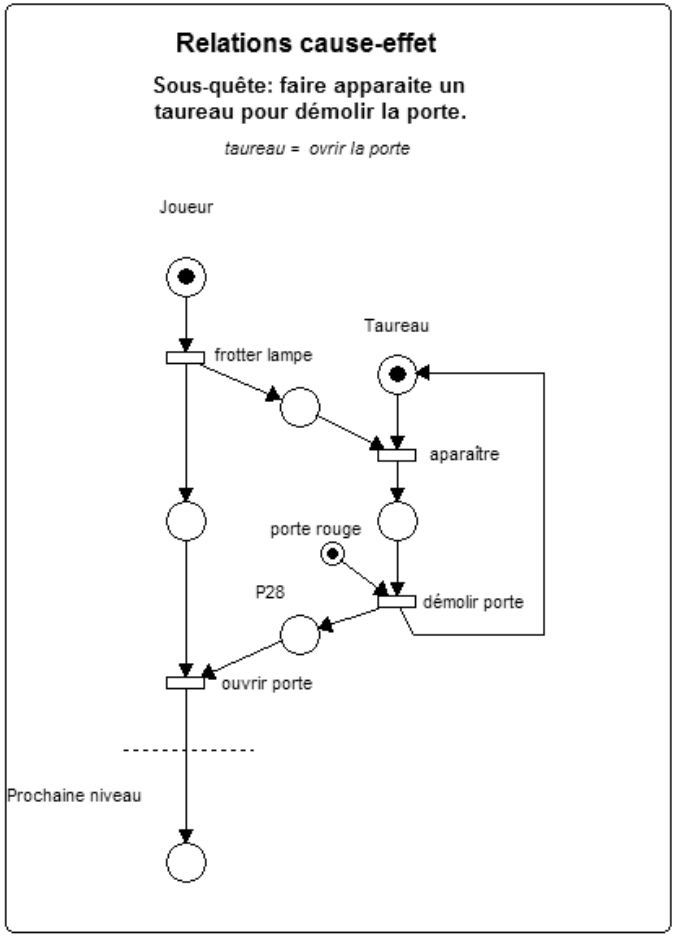
$M' = M - C^-(.,j) +$

Evolution du Marquage



$M' = M - C^{-}(.,j) + C^{+}(.,j)$

Exemple 1


Relations cause-effet
Sous-quête: faire apparaite un taureau pour démolir la porte.
taureau = ovrir la porte
Joueur
Taureau
frotter lampe
aparaître
porte rouge
P28
démolir porte
ouvrir porte
Prochaine niveau

Accessibilité

Soit R un réseau de Pétri M et M' deux marquages

M' est un marquage conséquent de M si
Il existe une transition T_j telle que
T_j soit franchissable à partir de M:
$M = C^-(.,j)$
et M' soit le marquage atteint par le franchissement de T_j

$M' = M + C^+(.,j) - C^-(.,j) = M + C(.,j)$

M' est accessible à partir de M
Il existe une suite de Marquages $M, M_1, \ldots M_{n-1}, M'$
et une suite de transitions $S = (T_{s1}, T_{s2}, \ldots, T_{sn})$
Tel que M_1 est un marquage conséquent de M par tir de T_{s1}
M_2 est un marquage conséquent de M_1 par tir de T_{s2}
…
M' est un marquage conséquent de M_{n-1} par tir de T_{sn}

Graphe des Marquages conséquents

Soit R un réseau de Pétri et M_0 un marquage initial de R

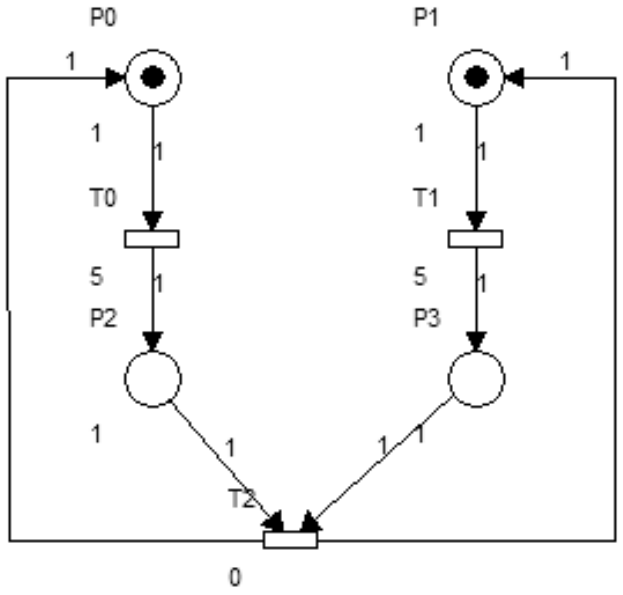
Le Graphe des marquages conséquent est G(X,U)

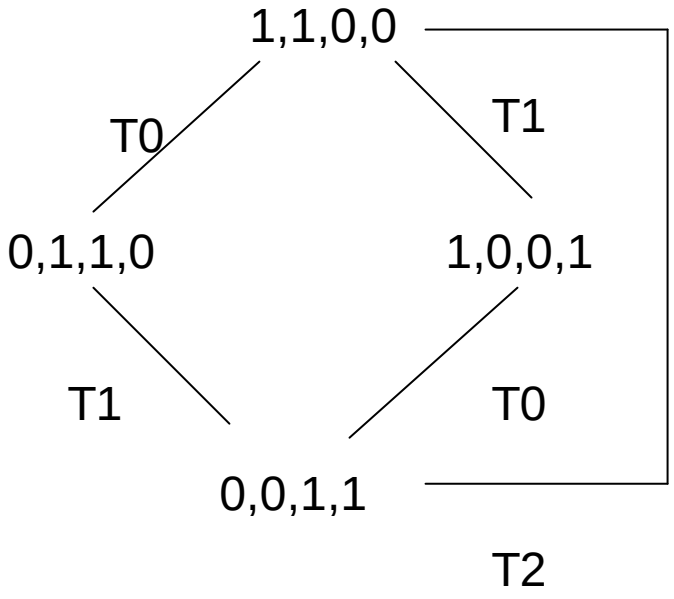
X={M accessibles à partir de M_0}

U={(M,M') tel que M et M' sont dans X et M' est conséquent de M}

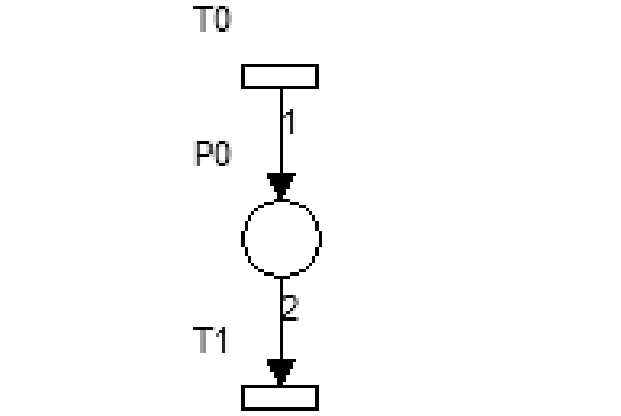
(M,M') est labellé avec le nom de la transition tirée

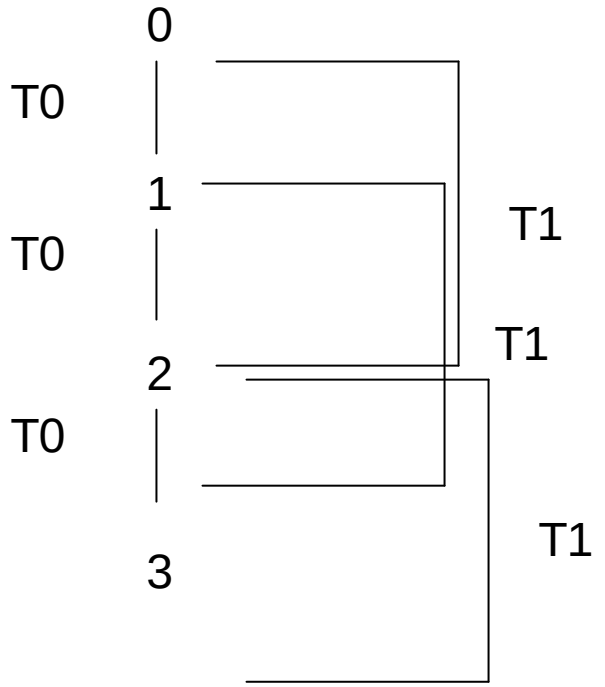
Exemple 2

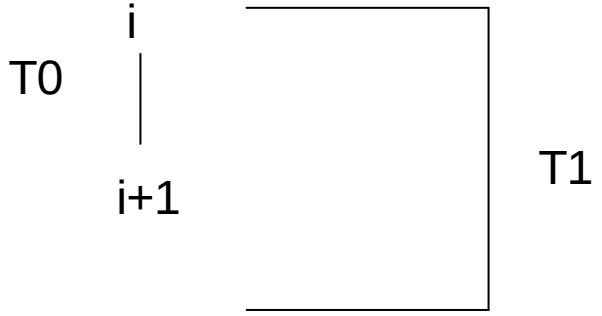

P0
P1
1
1
1
1
1
1
T0
T1
5
5
1
1
P2
P3
1
1
1
1
T2
0


1,1,0,0
T0
T1
0,1,1,0
1,0,0,1
T1
T0
0,0,1,1
T2

Exemple 3
un réseau non borné







Equation de franchissement

Soit R un réseau de Pétri M et M’ deux marquages tel M’ est accessible à partir de M

par une suite de transitions $S=(T_{s1}, T_{s2},\ldots,T_{sn})$

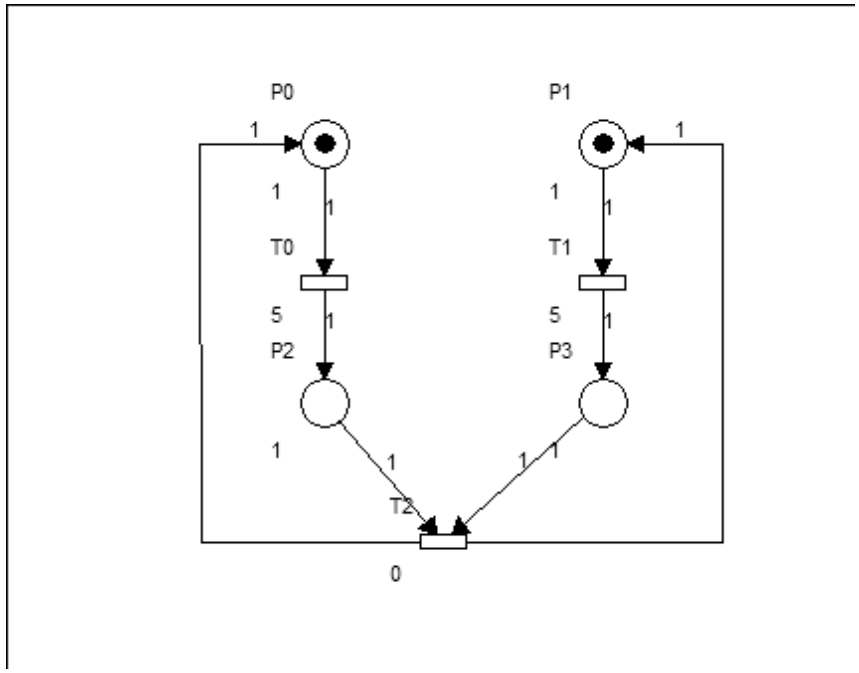
Notons s le vecteur colonne de $N^{card(T)}$ tel que

s(j)= nombre d’occurrence de T_j dans S

Alors

$$M'=M+C.s$$

Exemple



1,1,0,0

T0

1,0,1,0

T1

0,0,1,1

T2

1,1,0,0

T1

1,0,0,1

(1,0,0,1)=(1,1,0,0)+

-1	0	1
0	-1	1
1	0	-1
0	1	-1

$$\begin{pmatrix} 1 \\ 2 \\ 1 \end{pmatrix}$$

Invariants de Places (p semi flots)

Soit f un vecteur ligne de $N^{card(P)}$ tel que

f.C=0

Alors pour tout marquage acessible M à partir de M_0

$f.M=f.M_0$

De plus si f est positif ou nul et f(j) est positif

P_j est bornée pour tout marquage initial

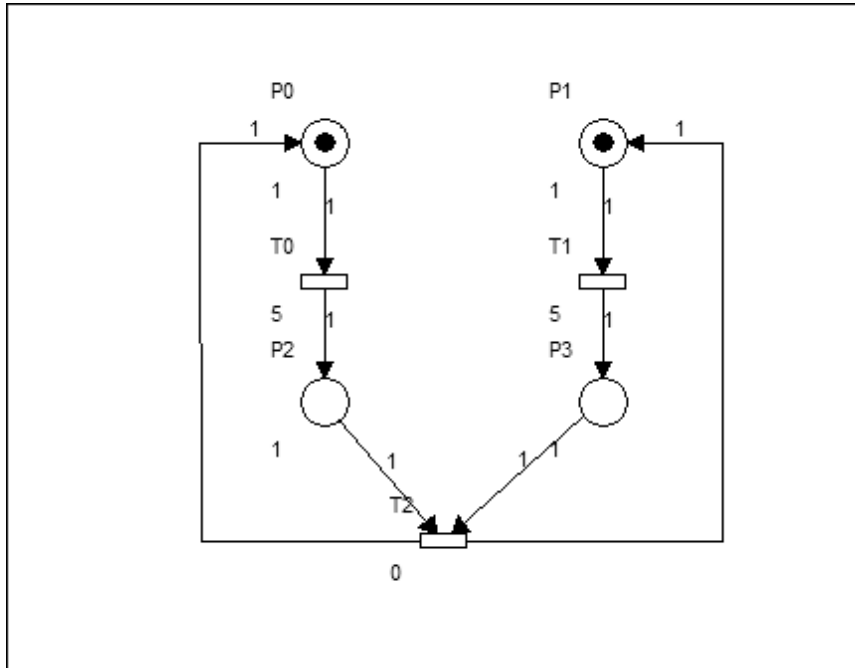
En effet $M= M_0 +C.s$ pour une séquence S donnée

Donc $f.M= f.M_0 +f.C.s= f.M_0$

Si f est positif ou nul et f(j) positif

$M(j) =f.M_0 /f(j)$

Exemple



(f1,f2,f3,f4)

-1	0	1
0	-1	1
1	0	-1
0	1	-1

=(0,0,0)

f0=f2
f1=f3
Don
M(0)+M(2)=1
Et
M(1)+M(3)=1

Invariants de Places (t semi flots)

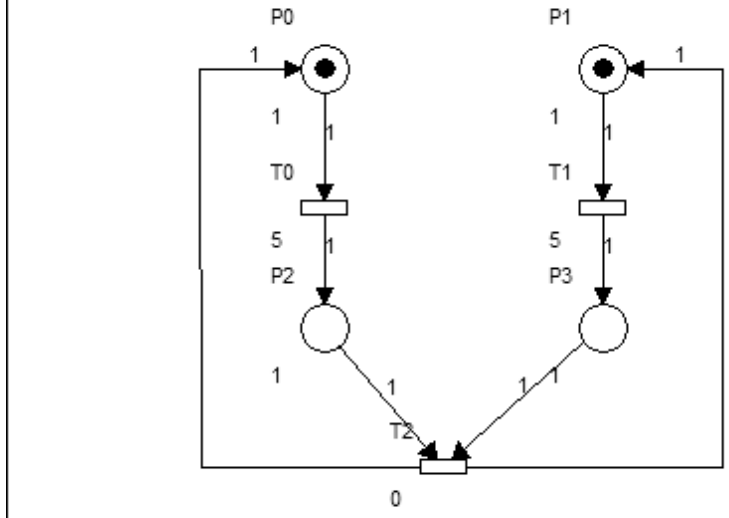
Soit S une séquence menant de M à partir de M_0 alors il existe un vecteur colonne s de $N^{card(T)}$ tel que

$M= M_0 +C.s$

Si il existe une séquence S menant de M à M alors il existe une solution entière et positive non nulle de l'équation

$$C.x=0$$

Exemple



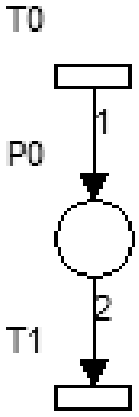
-1	0	1
0	-1	1
1	0	-1
0	1	-1

x1
x2
x3

=(0,0,0,0)

X1=X2=X3

Exemple 2
un réseau non borné



Pas d’invariant de places

Invariants de transition
2.X0=X1

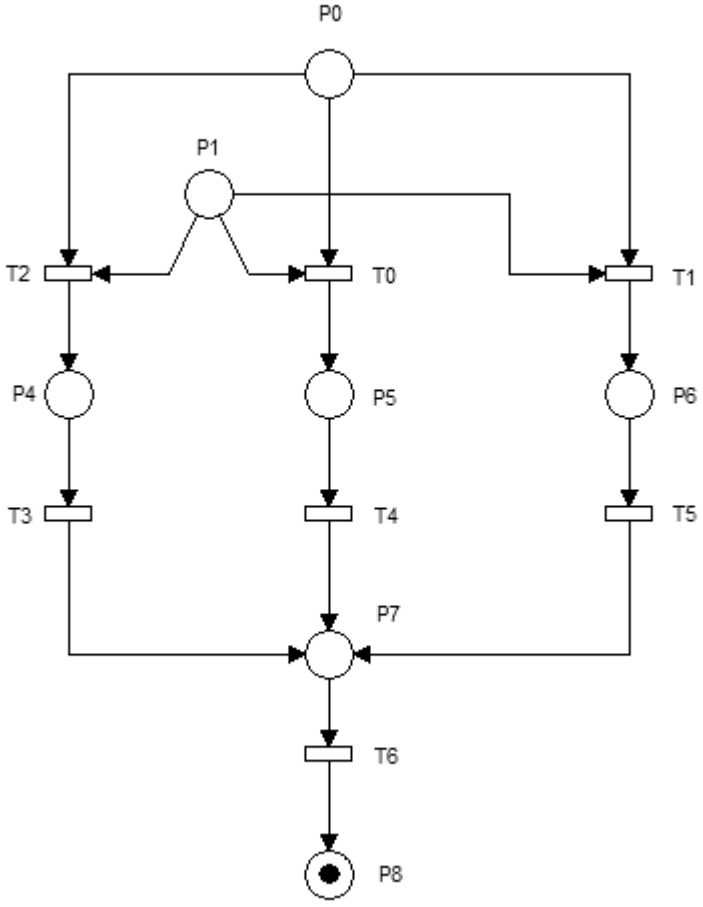
Utilisation dans les jeux

Un modèle global d’un scénario non linéaire:

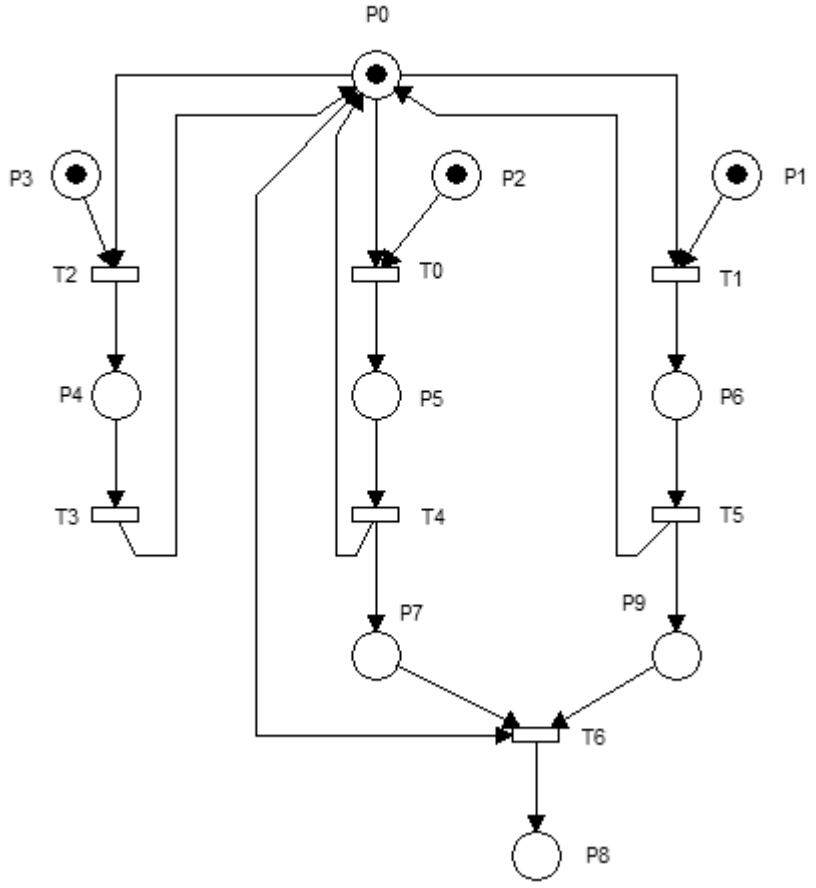
- Quêtes obligatoires, quêtes facultatives
- Niveau

Un découpage du logiciel avec identification des variables d’état

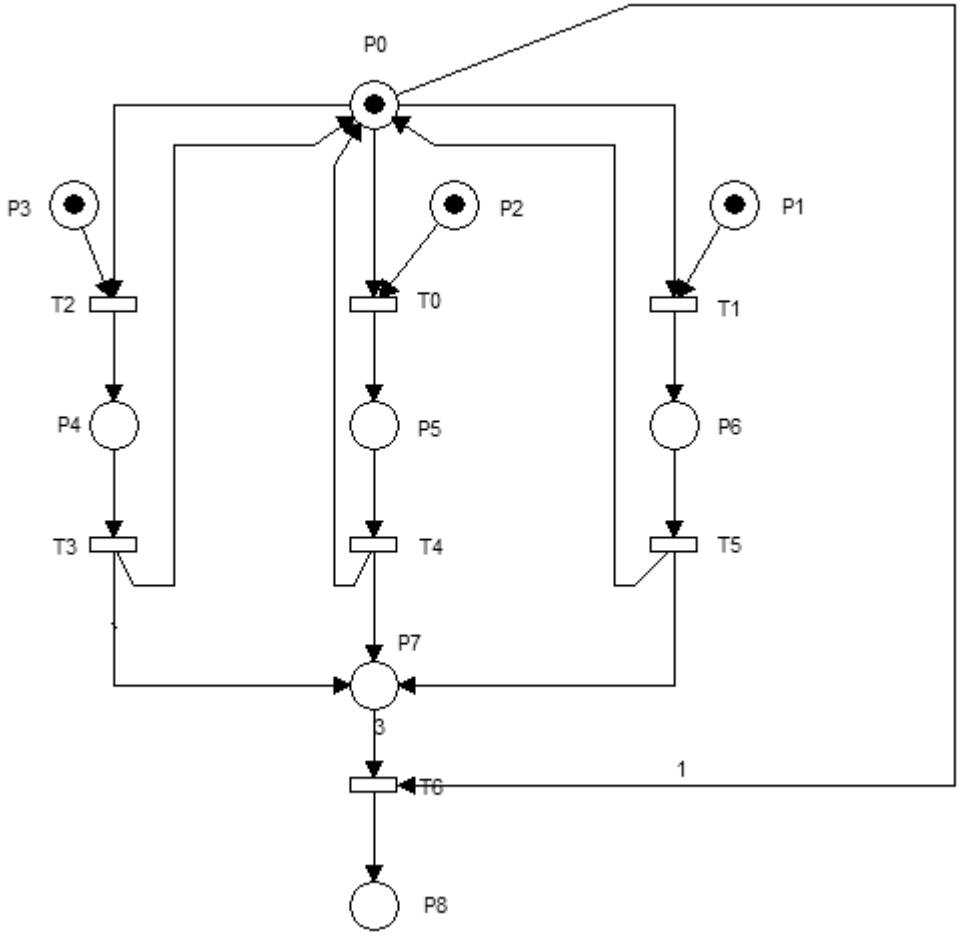
~~Faire 1 parmi 3~~



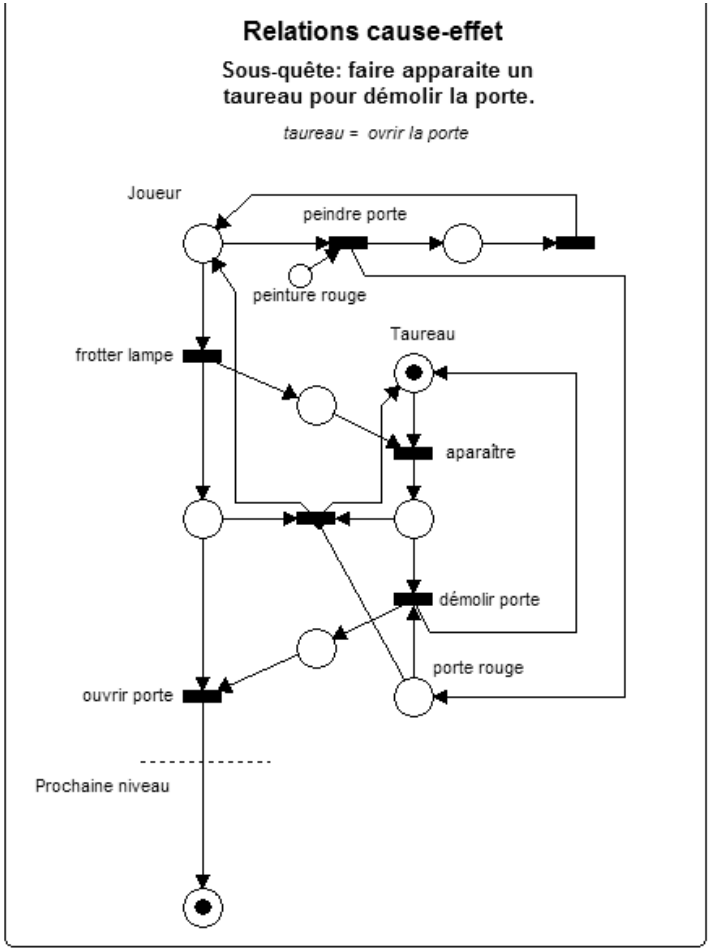
Faire 2 parmi 3



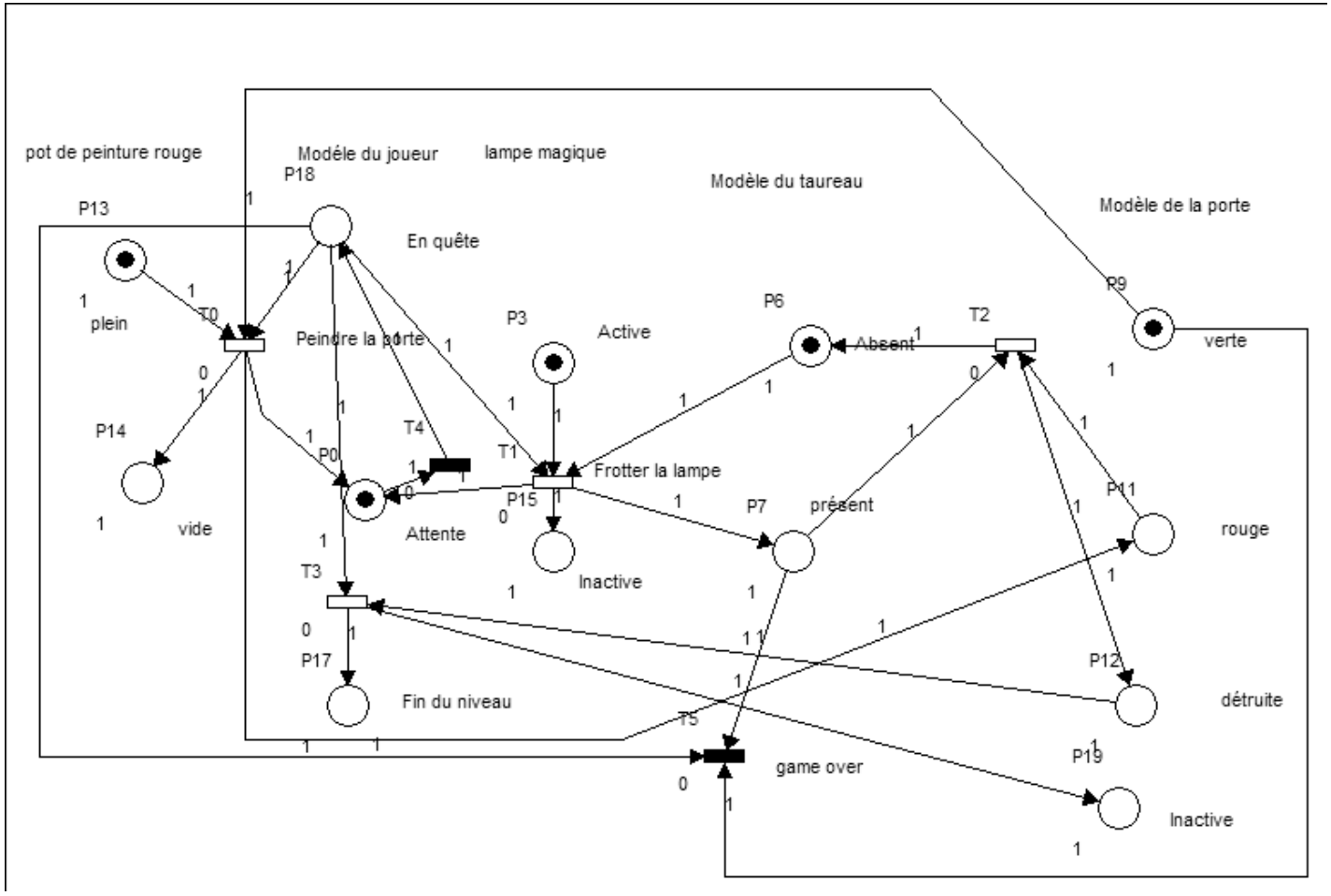
Faire 3 parmi 3



Modèle d’IA globlale



Modèle d'IA locale



Utilité et limite

- Ne pas essayer de tout « voir »
- Mieux qu’un graphe d’état global
- Utiliser des simulateurs (voir outils UML)
- Avec des automates simples pas de réel processus d’apprentissage (passer aux agents intelligents ou aux réseaux de neurones!!!)

Quelques idées d’architecture

- Distinguer: Le jeu à une IA globale ou des IA « locales »
- IA globale (RTS) Un automate global par acteur+ des automates locaux pour le comportement d’objets typés
- IA locale (jeu d’aventure): Automate par objets typés et localité
- Penser aux objets d’ambiance et les caméras
- Quand un jeu peut être joué en multijoueur ou contre la machine identifier l’automate machine joueur qui a le même langage de sortie que les entrées d’un joueur

L’espace d’état

- Prévoir un état de référence par défaut pour tout automate
- Construire un automate qui engrange l’état (par message recu des autres) et réinitialise le jeu
- Hiérachiser les points de reprise par niveau de jeu et localité: un niveau de reprise réinitialise toutes les variables de niveau inférieur à leurs valeurs par défaut