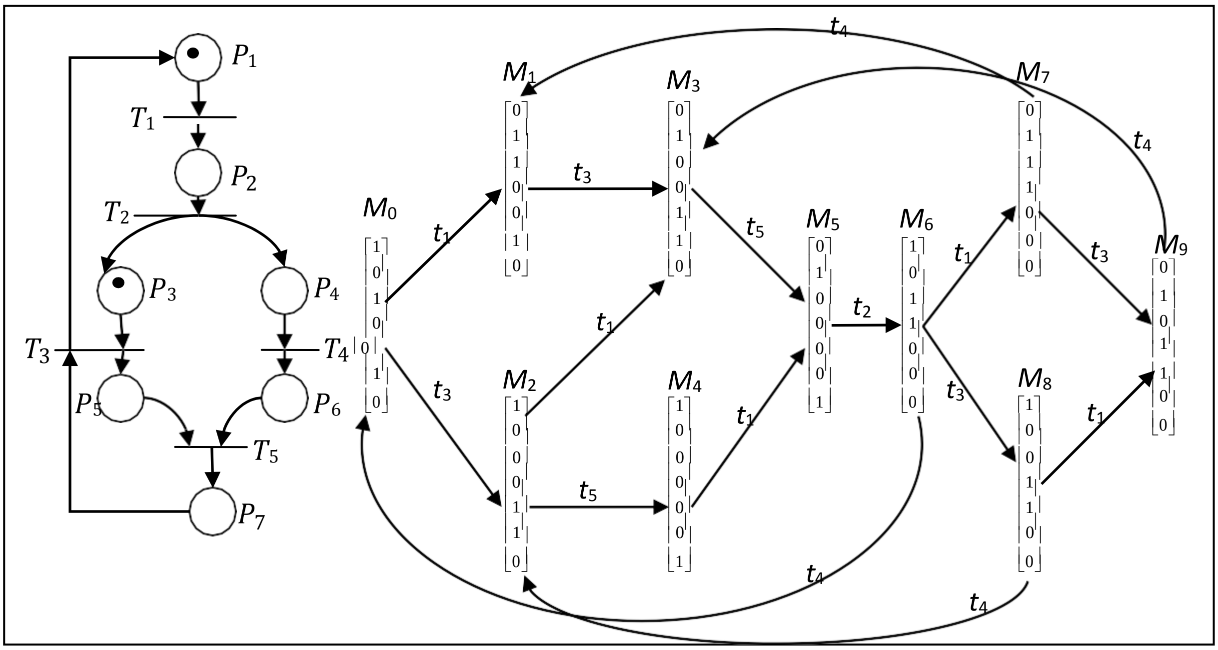
Chapitre 3

Analyse des systèmes avec prise en compte généralisé des dépendances



4.1 Introduction

Carl Adam Petri est un mathématicien allemand qui a défini un outil mathématique très général permettant de décrire des relations existant entre des conditions et des événements, de modéliser le comportement de systèmes dynamiques à événements discrets.

- Début des travaux 1960-1962 : ont donné lieu à de nombreuses recherches.
- 1972-1973, utilisation de cet outil pour la description d'automatismes logiques, ce qui a débouche sur le Grafcet. Cet outil permet l'analyse qualitative.

Il existe différents types de réseaux de Petri : temporisés, interprètes, stochastiques, colorés, continus et hybrides.

4.2 Les notions de base de réseaux de Petri

Un Réseau de Petri est un graphe orienté comportant :

- Un ensemble fini de places, $P = \{P_1, P_2, P_3, ..., P_m\}$, symbolisées par des cercles et représentant des conditions :

  Une ressource du système (ex. : une machine, un stock, un convoyeur, …)

  L'état d'une ressource du système (ex. : machine libre, stock vide, convoyeur en panne, ……etc)

  Un ensemble fini de transitions, $T = \{T_1, T_2, T_3, ..., T_n\}$, symbolisées par des traits et représentant l'ensemble des événements (les actions se déroulant dans le système) dont l'occurrence provoque la modification de l'état du système.

- Un ensemble fini d'événements associés à chaque transition.
- Un ensemble fini d'arcs orientés qui assurent la liaison d'une place vers une transition ou d'une transition vers une place.

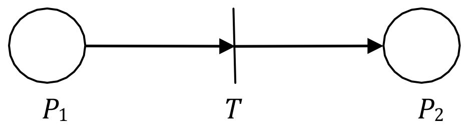


Figure 4.1 : Graphe orienté

Définition: Un RdP synchronisé est défini par un 5-tuplet $R = \{P, T, Pré, Post, M_0\}$ où

$P = \{P_1, P_2, P_3, \ldots, P_N\}$ est l'ensemble des places ;

$T = \{T_1, T_2, T_3, \ldots, T_L\}$ est l'ensemble des transitions ;

Entrée (ou Pré) est une application, Entrée: P x T → N, appelée application d'incidence avant.

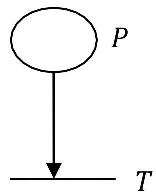


Figure 4.2 : Application d'incidence avant

Sortie (ou Post) est une application, Sortie: P x T → N, appelée application d'incidence arrière.

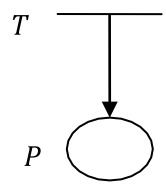
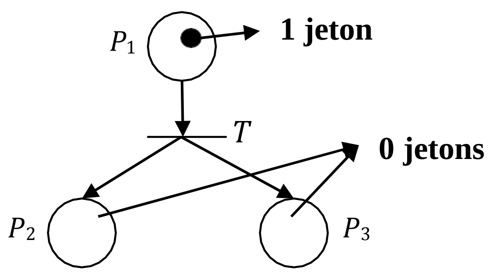


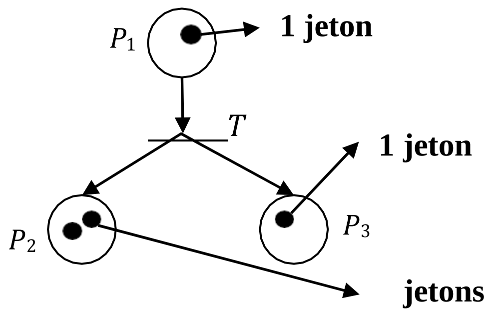
Figure 4.3 : Application d'incidence arrière

M_0 est l'état initial $P \rightarrow R$.

Exemple 4.1



$$M_0 = P_1P_2P_3 \Rightarrow M_0 = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$$



$$M_0 = P_1P_2P_3 \Rightarrow M_0 = \begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix}$$

Figure 4.4 : Réseau de Petri marqué

Remarque 4.1 : Un jeton peut avoir plusieurs significations en fonction de la place dans lequel il se trouve.

Exemple 4.2

P_1 représente un stock : le nombre de jetons en P_1 indique le nombre de pièces stockées

- P_2 représente une machine en cours de traitement : un jeton en P_2 indique que la machine traite une pièce.
- P_3 représente une machine libre : un jeton en P_3 indique que la machine est libre.

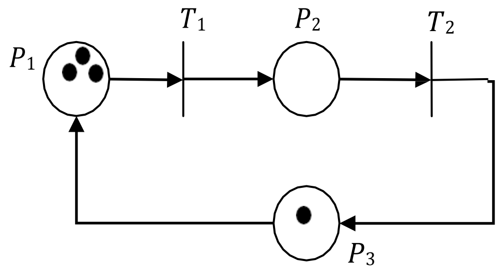


Figure 4.5 : RDP d'un stock- machine

Exemple 4.3

La figure 4.6 présente un atelier constitué d'une machine de coupe et d'un stock. Quand une commande arrive et que la machine de coupe est disponible, la commande peut être traitée (Opération de découpe). Une fois le traitement terminé, la commande qui a été traitée est stockée. Sinon, la commande doit attendre que la machine se libère avant de pouvoir être traitée.

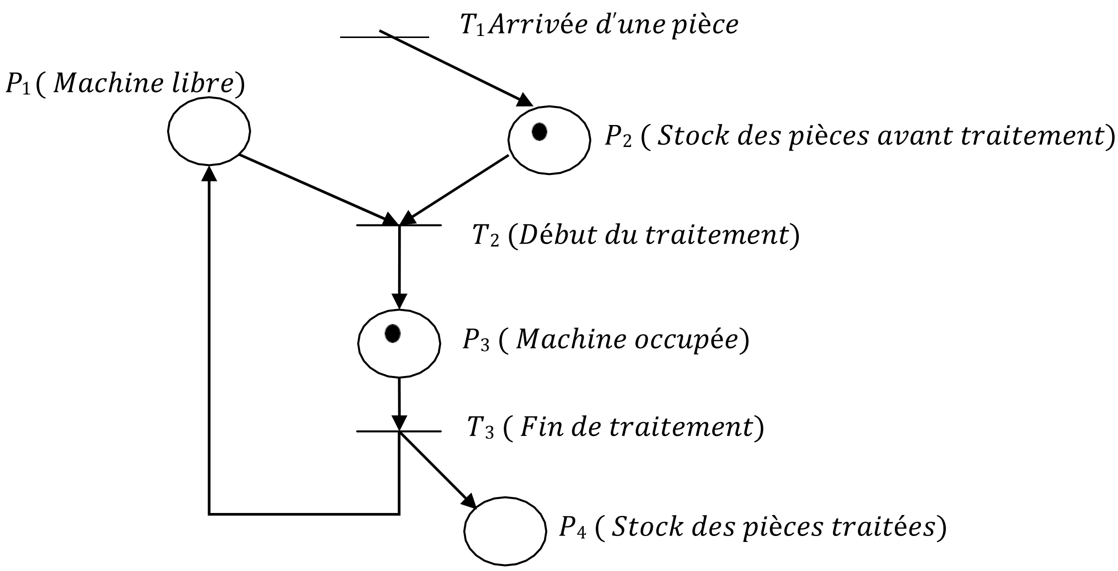


Figure. 4.6 : Modélisation d'un atelier de coupe de bois

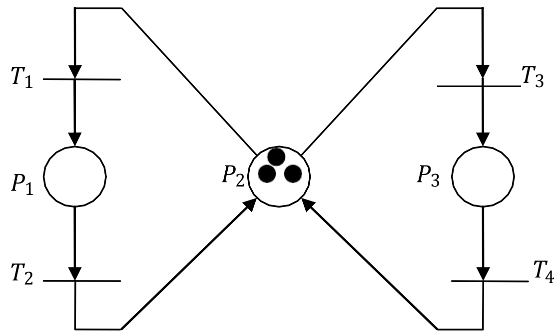
Au lieu de *Pré* et *Post* en général nous utilisons *W*, la matrice d'incidence qui est calculable à partir de l'ensemble *Pré* et *Post* comme ci-dessous :

$W = Post - Pré$

Exercice 4.1

Pour le RdP ci-contre :

- Indiquer le marquage initial
- Etablir la matrice d'entrée Pré
- Etablir la matrice de sortie Post
- Etablir la matrice d'incidence W



Solution

$$M_0 = \begin{bmatrix} 0 \\ 3 \\ 0 \end{bmatrix}$$

$$Pré = \begin{bmatrix} 0100 \\ 1010 \\ 0001 \end{bmatrix} \quad Post = \begin{bmatrix} 1000 \\ 0101 \\ 0010 \end{bmatrix} \quad W = \begin{bmatrix} 1 & -1 & 0 & 0 \\ -1 & 1 & -1 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

Remarque 4.2 : Pour faciliter le processus de remplissage des matrices, vous pouvez utiliser les tableaux illustrés comme suit :

-

+

Pré	T_1	T_2	T_3	T_4
P_1	***0***	***1***		
P_2	***1***			
P_3	***0***			

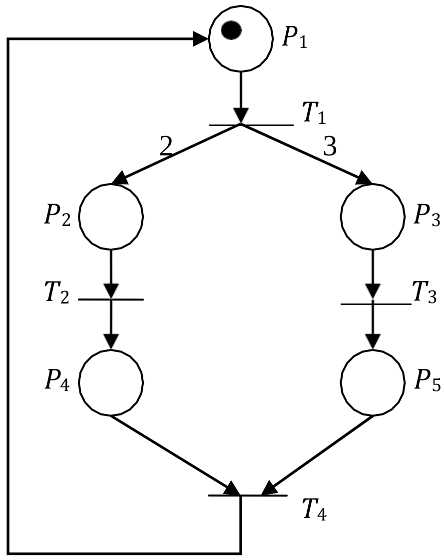
Post	T_1	T_2	T_3	T_4
P_1	***1***			
P_2	***0***			
P_3	***0***			

W	T_1	T_2	T_3	T_4
P_1	***1***			
P_2	***-1***			
P_3	***0***			

Tableau 4.1 : Processus de remplissage des matrices

Exercice 4.2

Pour le réseau de Petri suivante :



1- Etablir la matrice d'incidence avant

2- Etablir la matrice d'incidence arrière

3- Etablir la matrice d'incidence W

Solution

$$Pré = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad Post = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 2 & 0 & 0 & 0 \\ 3 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \quad W = \begin{bmatrix} -1 & 0 & 0 & 1 \\ 2 & -1 & 0 & 0 \\ 3 & 0 & -1 & 0 \\ 0 & 1 & 0 & -1 \\ 0 & 0 & 1 & -1 \end{bmatrix}$$

Remarque 4.3 : Chaque arc possède un entier positif qui représente son poids. Par convention, lorsque le poids n'est pas précisé sur un arc, alors ce poids vaut 1.

4.3 Propriétés des réseaux de Petri

Dans cette section, on considère les réseaux de Petri ordinaires et nous présentons les propriétés qui nous seront nécessaires pour la présentation de notre contribution.

Marquages accessibles : On notera M_R l'ensemble des *marquages accessibles* d'un RdP R à partir du marquage initial M_0.

RdP borné : Une place P_i est dit *bornée* pour un marquage initial M_0 s'il existe un entier naturel k, tel que pour tout marquage accessible à partir de M_0, le nombre de marques dans P_i est inférieur ou égal à k.

Transition vivante : Une transition T_j est *vivante* pour un marquage initial M_0 si pour tout marquage accessible $M_i \in \mathrm{M}_R$, il existe une séquence de franchissement S qui contient la transition T_j, à partir de M_i.

RdP vivant : Un RdP est *vivant* pour un marquage initial M_0 si toutes ses transitions sont vivantes pour M_0.

Blocage : un *blocage* est un marquage tel qu'aucune transition ne peut être validée.

Conflit structurel : Un *conflit structurel* correspond à un ensemble d'au moins 2 transitions T_1 et T_2 qui a une place d'entrée en commun P_i.

Conflit effectif dans un RdP sauf : Il y a un conflit effectif dans un RdP *sauf* lorsqu'il y a au moins deux transitions validées synchronisées avec le même événement.

Invariants : Soit R un RdP et P l'ensemble de ses places. On a un *invariant de marquage,* s'il existe un sous-ensemble de places P' inclus dans P et un vecteur de pondération Q $=(q_1, q_2,\dots, q_r)$, dont tous les poids q_i sont des nombres entiers positifs, tels que :

$$q_1 M(P_1) + q_2 M(P_2)+\dots+ q_r M(P_r) = \text{constante, pour tout } M \in \mathrm{M}_R.$$

4.4 Franchissement d'une transition

Le franchissement d'une transition ne peut s'effectuer que si chacune des places en amont de cette transition contient au moins une marque. On dit alors que la transition est *franchissable* ou *validée*. Le franchissement d'une transition fait évoluer le marquage du réseau en retirant un marque de chacune des places en amont de la transition et en ajoutant une marque dans chacune des places en aval de la transition. Pour les RdP saufs, il y a un seul franchissement à la fois, et la durée de ce franchissement est nulle.

Dans un RdP synchronisé, une transition validée est franchie à l'occurrence de l'événement qui lui est associé.

Dans la figure 4.2-*a* la transition T_1 est franchissable lorsque l'événement σ_1 se produit. Le résultat du franchissement de cette transition est indiqué dans la figure 4.2-b.

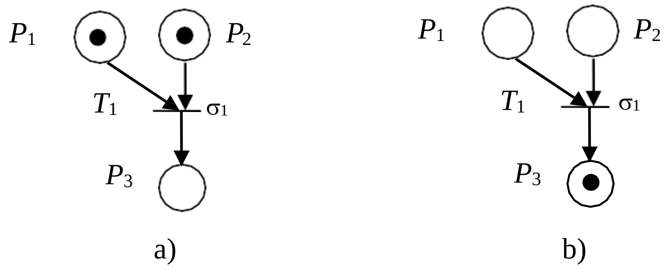


Figure. 4.7 : Franchissement d'une transition

A chaque séquence de franchissement, est associé un vecteur caractéristique noté $\bar{s}$. C'est un vecteur de dimension L (nombre de transitions) où le composant numéro j correspond au nombre de franchissements de la transition T_j dans la séquence S. Si la séquence de franchissement S est réalisable à partir d'un marquage M_i, le marquage atteint M_k est donné par l'équation fondamentale :

$$M_k = M_i + W.\,\bar{s}$$

$\bar{s}$ est le vecteur caractéristique d'une séquence S qui mène de M_i à M_k.

4.5 Graph des marquages atteignables et Graph de couvrement

Pour connaître le comportement d'un réseau, l'idée la plus simple serait de construire le graph des marquages atteignables. Dans l'arbre des marquages atteignables, chaque nœud correspond à un marquage atteignable, et l'arc reliant deux nœuds correspond au franchissement d'une transition qui conduit d'un marquage à l'autre. L'arbre des marquages atteignables est un moyen idéal pour vérifier les propriétés comme la vivacité et l'atteignabilité pour les réseaux bornés. Mais le problème se pose quand un réseau de Petri n'est pas borné, puisque le nombre de marquages atteignables devient alors infini, et le nombre de nœuds dans l'arbre des marquages atteignables est également infini. L'idée est donc de construire un arbre de marquages ayant un nombre fini de nœuds que l'on appelle graphe de couvrement.

Dans ce graphe, nous introduisons un symbole $\boldsymbol{\omega}$ pour indiquer que le nombre de jetons dans une place est "infini". Ce marquage restera ω dans tous les développements suivants et le premier point précédent s'applique aussi aux marquages contenant ce symbole.

Un nœud correspondant à un marquage tel qu'aucune transition n'est tirable sera marqué « **dead-end** » et constituera une feuille de l'arborescence, Tous les nœuds qui ne sont pas marqués « **old** » et qui admettent aux moins un descendant sont marqués « **new** ».

Exemple 4.4 : graphe de marquage atteignable

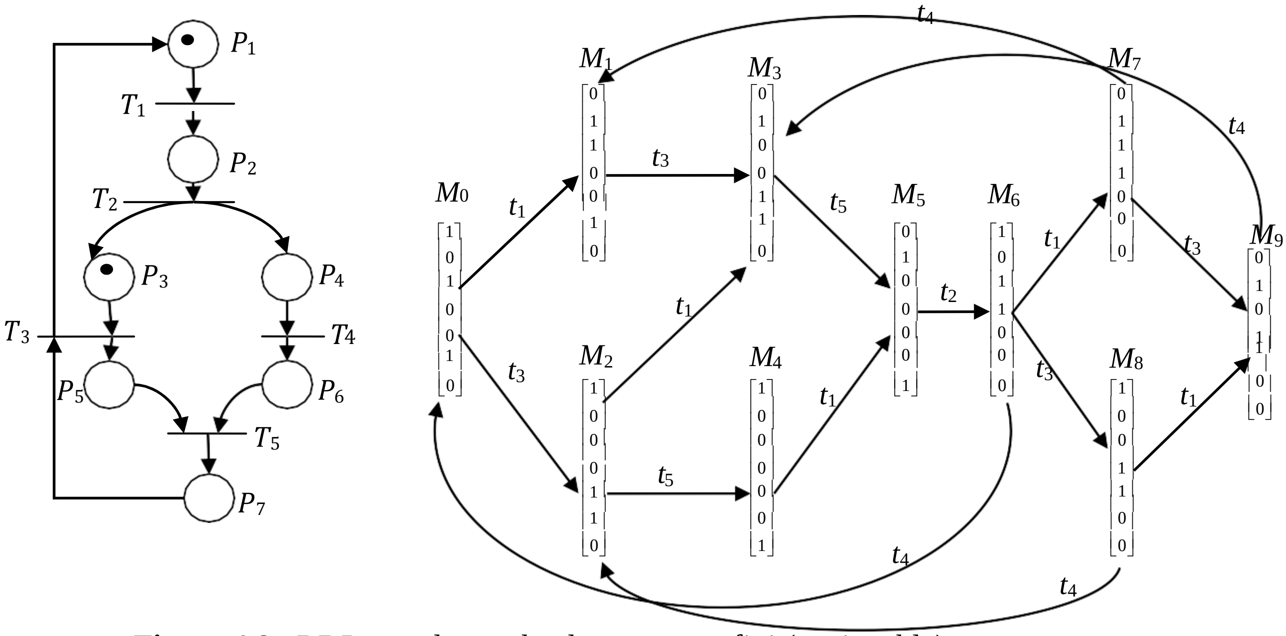


Figure 4.8 : RDP avec le graphe de marquage fini (atteignable)

Exemple 4.5 : graphe de couvrement

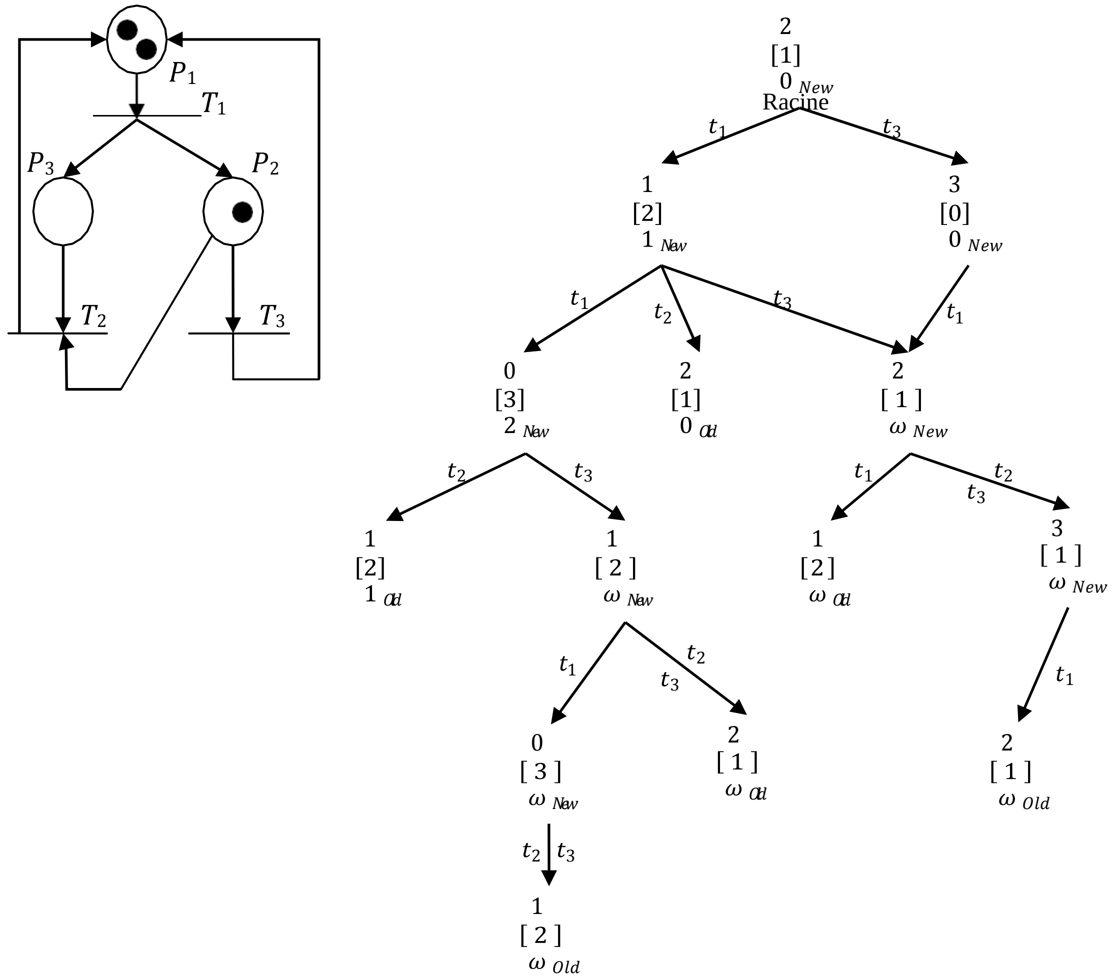
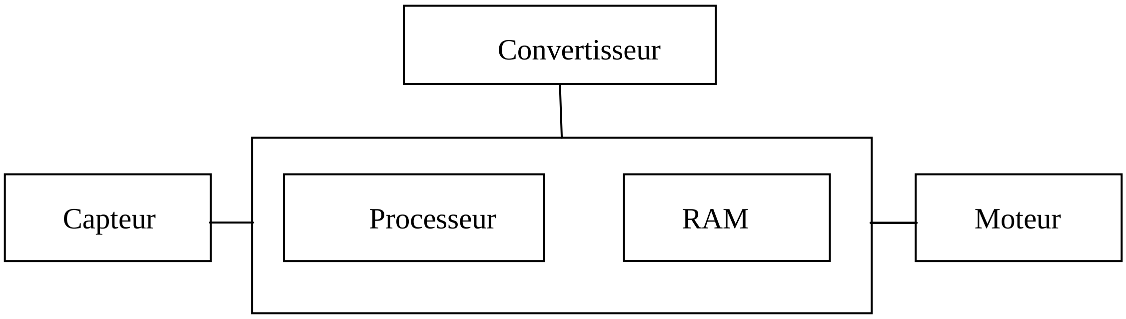


Figure 4.9 : RDP avec le graphe de couvrement

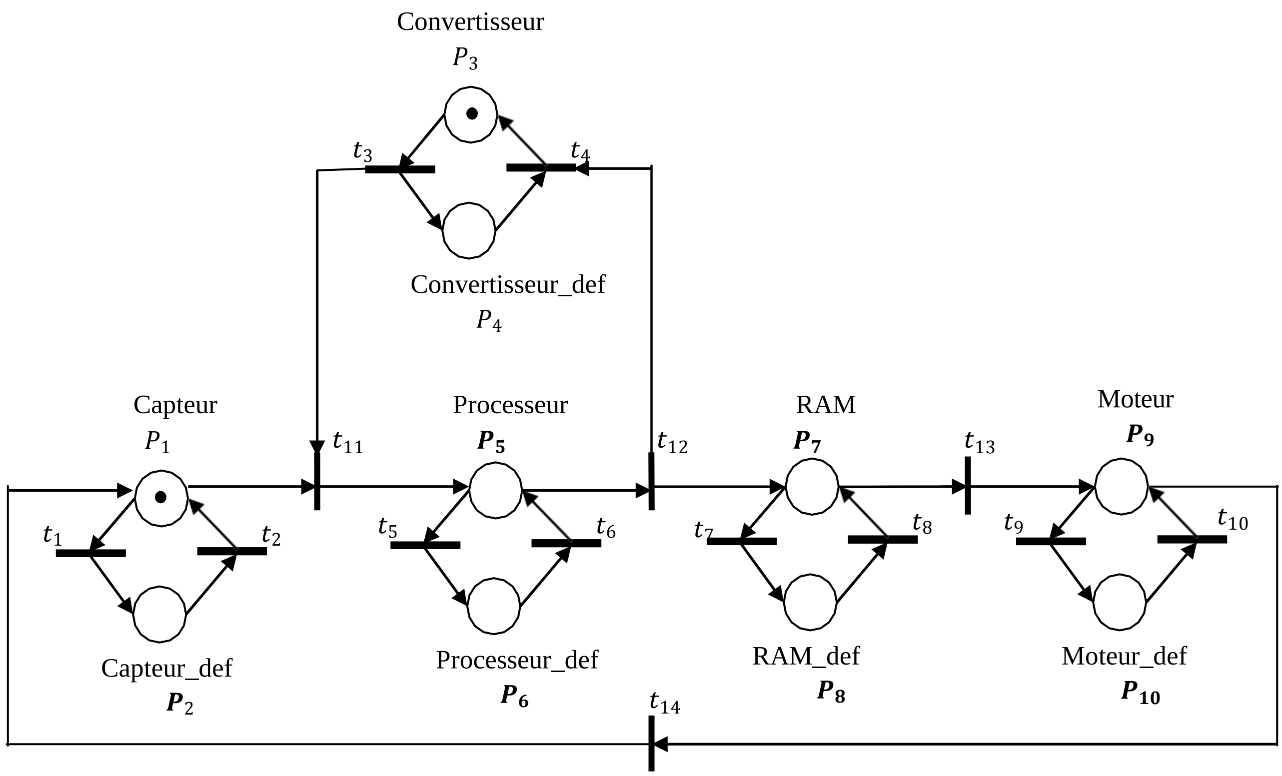
Remarque 4.4 : Dans le cas d'un graphe des marquages de dimension infinie, le graphe de couverture ne permet pas d'analyser la vivacité. L'introduction du symbole ω correspond à une perte d'information.

Exercice 4.3

La figure ci-dessous représente l'ouverture automatique d'une portière : le conducteur du véhicule en possession de la clé est détecte par le capteur du système ce qui déclenche l'activation d'un convertisseur qui alimente le reste du système. L'information est reçue puis traitée par un processeur muni d'un RAM qui déclenche un moteur qui ouvrira la portière.



Le système en un réseau de Petri : chaque composant a deux états correct ou défaillance, tout équipement arrêté ne peut tomber en panne. Le système sous forme de réseau de Petri est modélisé comme suit :



1- Etablir la matrice d'incidence avant
2- Etablir la matrice d'incidence arrière
3- Etablir la matrice d'incidence W
4- Etablir le graphe de marquage.

Solution

1- Matrice d'incidence avant

$$Pré = \begin{bmatrix} 1&0&0&0&0&0&0&0&0&0&1&0&0&0 \\ 0&1&0&0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&1&0&0&0&0&0&0&0&1&0&0&0 \\ 0&0&0&1&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&1&0&0&0&0&0&0&1&0&0 \\ 0&0&0&0&0&1&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&1&0&0&0&0&0&1&0 \\ 0&0&0&0&0&0&0&1&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&1&0&0&0&0&1 \\ 0&0&0&0&0&0&0&0&0&1&0&0&0&0 \end{bmatrix}$$

2- Matrice d'incidence arrière

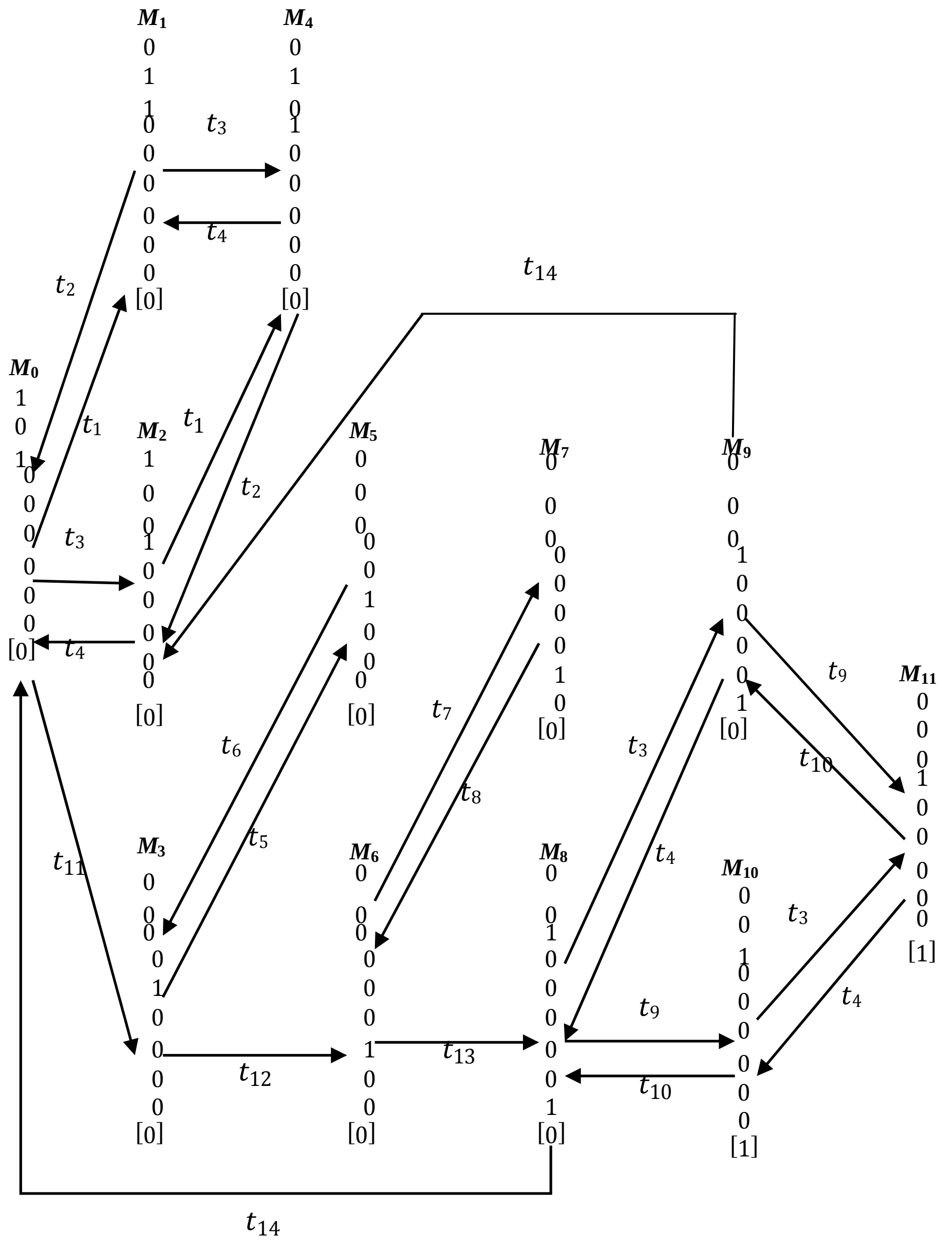
$$Post = \begin{bmatrix} 0&1&0&0&0&0&0&0&0&0&0&0&0&1 \\ 1&0&0&0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&1&0&0&0&0&0&0&0&0&1&0 \\ 0&0&1&0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&1&0&0&0&0&1&0&0&0 \\ 0&0&0&0&1&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&1&0&0&0&1&0&0 \\ 0&0&0&0&0&0&1&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&1&0&0&1&0 \\ 0&0&0&0&0&0&0&0&1&0&0&0&0&0 \end{bmatrix}$$

3- Matrice d'incidence W

W=Post-Pré

$$W = \begin{bmatrix} -1&1&0&0&0&0&0&0&0&0&-1&0&0&1 \\ 1&-1&0&0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&-1&1&0&0&0&0&0&0&-1&0&1&0 \\ 0&0&1&-1&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&-1&1&0&0&0&0&1&-1&0&0 \\ 0&0&0&0&1&-1&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&-1&1&0&0&0&1&-1&0 \\ 0&0&0&0&0&0&1&-1&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&-1&1&0&0&1&-1 \\ 0&0&0&0&0&0&0&0&1&-1&0&0&0&0 \end{bmatrix}$$

4- Graphe de marquage



4.6 RdP Stochastiques

Les RdP Stochastiques ont été introduits pour répondre à des problèmes liés à la sûreté de fonctionnement. Ces problèmes faisant intervenir des phénomènes aléatoires, on associe des temps de franchissement aléatoires, donc non déterministes distribués par une loi exponentielle aux transitions du réseau de Petri.

4.6.1 Définition des RdP Stochastiques

Un RdP stochastique est un 5-uplet RdPS = (P, T, E, μ, M_0).

P = {P_1, P_2,..., P_n}: ensemble fini et non vide de places. T = {T_1, T_2, ...T_m}: ensemble fini et nom vide de transitions. A chaque Ti on associe un taux de franchissement μ_i.

E = ensemble des arcs μ = {μ_1, μ_2, …,μ_n}: ensemble des taux de franchissements

M_0: Vecteur marquage initial.

4.7 Analyse d'un RdP stochastique

Pour analyser un RdP stochastique, deux approches complémentaires peuvent être utilisées, les processus de Markov et les propriétés de conservation d'un RdP. Pour la première approche, elle consiste à construire le graphe des marquages accessibles du RdP autonome sous-jacent et à étiqueter chaque arc par un taux de franchissement qui dépend du taux associé à la transition et du marquage des places en amont de cette transition.

Exemple 4.6

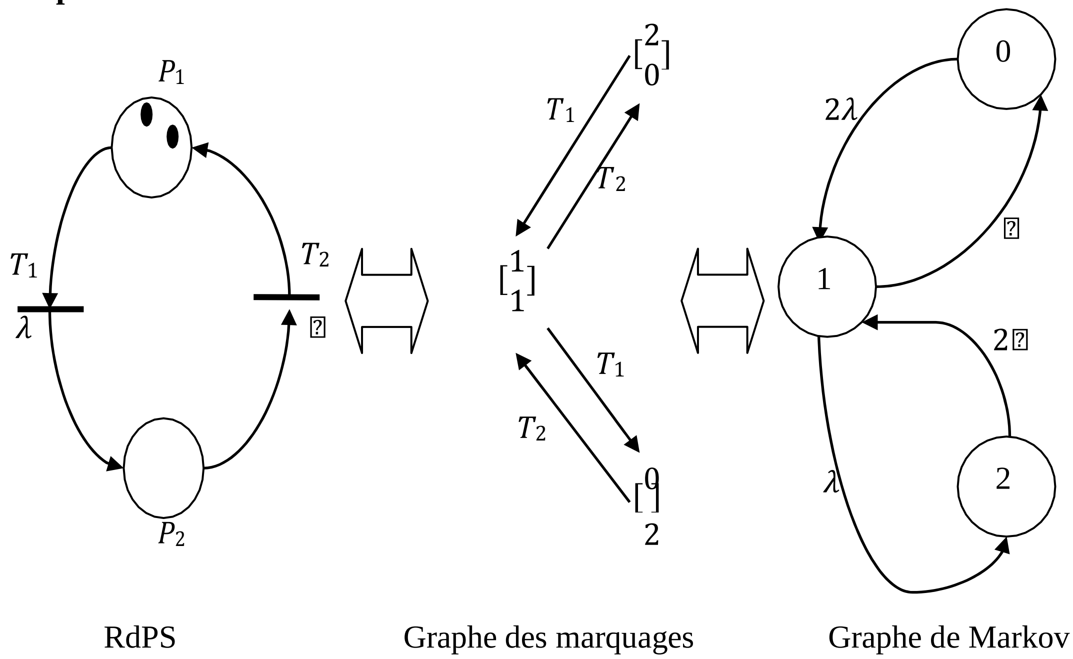


Figure 4.10 : Liaison entre réseau de réseau de Peri stochastique (RDPS) Chaine de Markov(GdM)

<u>Remarque 4.5</u> : L'exploitation quantitative d'un réseau de Petri se fait généralement, soit en traitant le modèle markovien issu du graphe des marquages accessibles du réseau de Petri, chaque marquage non évanescent correspondant à un état de la chaîne de Markov, soit en animant par simulation de Monte-Carlo directement le modèle réseau de Petri et non son graphe d'accessibilité.

4.8 Avantages et limites des RdP

4.8.1 Avantages

Le formalisme des RdP présente les avantages suivants

1- Définition formelle : cette caractéristique efface toute ambiguïté car chaque modèle possède une sémantique bien définie.
2- Les RdP sont exécutables : il existe des programmes qui permettent d'interpréter les modèles en RdP et de simuler le fonctionnement du système. Ceci permet une vision dynamique du système.
3- Expression puissante : les RdP sont adaptés pour décrire des comportements complexes, réactifs ou concurrents.
4- Représentation graphique : cette qualité facilite l'interprétation et la compréhension des modèles.

4.8.2 Limites

Malgré ces multiples qualités, les RdP soufrent de certains points négatifs :

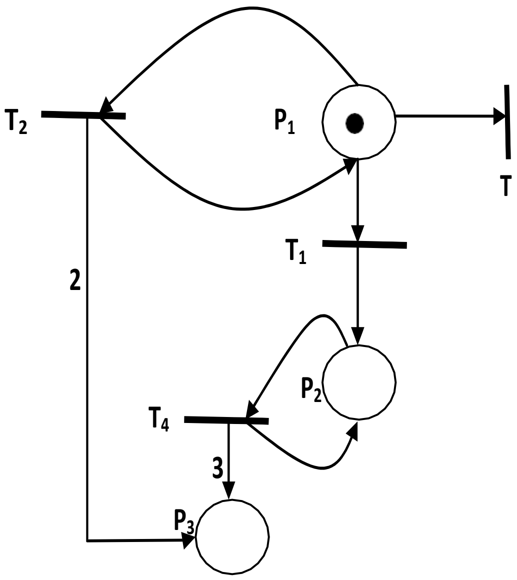
1- Manque de structuration : plus le système est complexe plus la taille du modèle produit est importante, et plus leur maîtrise est compliquée.
2- Structure de données : les réseaux places/transitions ne permettent pas de décrire la structure des données manipulées par le système.

4.9 Exercices proposés

Exercice P4.1

Pour le réseau de Petri ci-contre:

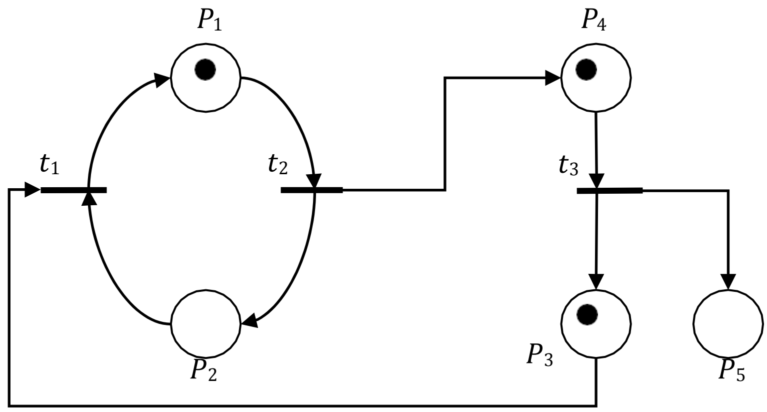
- Etablir la matrice d'incidence avant (Pré)
- Etablir la matrice d'incidence arrière (Post)
- Etablir la matrice d'incidence W
- Etablir le graphe de marquage



Chapitre 4 : *Analyse des systèmes avec prise en compte généralisé des dépendances*

Exercice P4.2

Soit le RdP suivant.



1) Déterminer la matrice d'incidence U.
2) Déterminer le marquage initial M_0
3) Considérer la séquence S = <t_2, t_1, t_3, t_3>. Déterminer le marquage obtenu après le tirage de la séquence S en utilisant l'équation d'état $M_k = M_i + W.\bar{s}$ où $\bar{s}$ est le vecteur caractéristique d'une séquence *S*.
4) Etablir le graphe de marquage

Exercice P4.3

Soit un système à deux équipements A_1 et A_2 identiques, assurant la même fonction et utilisés de façon simultanée. A_1 et A_2 ne sont pas réparables. Les pannes de A_1 et A_2 sont indépendantes c'est à dire que la panne de A_1 ou A_2 n'entraîne pas la perte de la fonction à assurer.

1- Modélisé le système par :

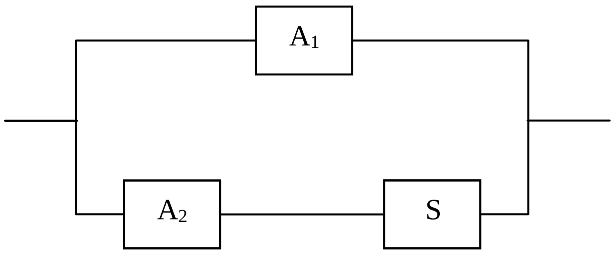
a) Arbre de cause ;

b) Réseau de Petri ;

c) Chaîne de Markov.

Exercice P4.4

La figure ci-dessous comporte deux composants A_1 et A2. Si on suppose qu'en cas de défaillance de A_1, il est nécessaire de commuter un switch S pour utiliser A_2, on peut dire que le dispositif A_2 ne sera disponible que si le dispositif S fonctionne.

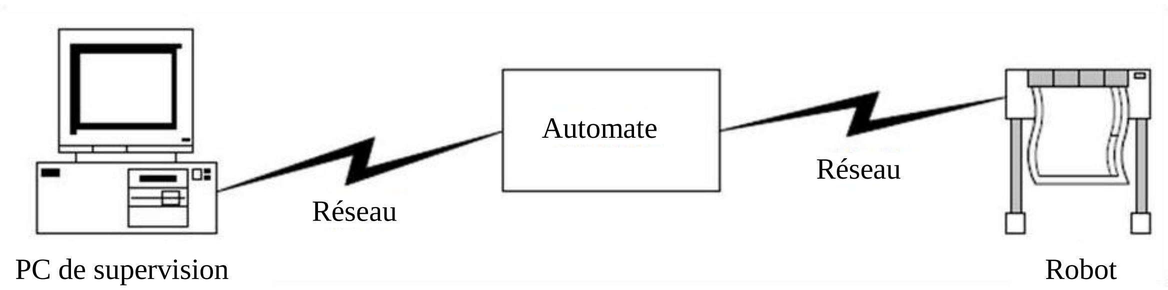


2- Modélisé le système par :

1) Arbre de cause ;
2) Réseau de Petri ;
3) Chaîne de Markov.

Exercice P4.5

Un système est composé d'un PC de supervision, d'un automate et d'un robot reliés par réseau.



Le PC de supervision possède un algorithme qui permet de déterminer les N prochaines actions du robot (N est fixe). L'automate reçoit, via le réseau, les N prochaines actions du robot et les envoie une par une au robot, qui les exécute. Lorsque le robot ne peut pas exécuter une action (panne…) il prévient l'automate, qui prévient le PC de supervision, qui décide alors d'arrêter le système. Lorsque le robot a terminé correctement une action, il prévient l'automate qui lui envoie la prochaine action. Lorsque l'automate envoie la dernière action au robot, il prévient le PC de supervision et lui demande les N prochaines actions. L'automate et le PC de supervision contrôlent que les échanges d'information par le réseau se sont bien déroulés. À l'état initial, le système est au repos. Pour arrêter le système, un message d'arrêt est émis par le PC de supervision en remplacement des N prochaines actions. Lorsque l'automate reçoit ce message, il met le robot au repos dès que ses dernières actions sont exécutées. Déterminer à l'aide de réseaux de Petri communiquant le fonctionnement du système (un RdP par entité du système : Robot, Automate, PC de supervision).