

Les Réseaux de Petri



Robert Valette

LAAS-CNRS Toulouse,

<http://www.laas.fr/~robert>

ou directement : <http://www.laas.fr/~robert/enseignement.d>

e-mail robert@laas.fr

Septembre 2002

Chapitre 1

Un peu d'histoire.

Les réseaux de Petri sont un outil graphique et mathématique qui s'applique à un grand nombre de domaines où les notions d'événements et d'évolutions simultanées sont importantes.

Cette théorie est encore toute jeune puisqu'elle est née de la thèse, intitulée *Communication avec des Automates*, que Carl Adam Petri a présentée en 1962 à l'Université de Darmstadt. Carl Adam Petri est né en 1926 à Leipzig, il est professeur à l'Université de Bonn. Anatol W. Holt fut séduit par ce travail et sous son impulsion un groupe de chercheurs du *Massachusetts Institute of Technology* (MIT) jeta les bases, entre 1968 et 1976, de ce qui devint les *Réseaux de Petri*. Parmi ces pionniers on peut citer: F. Commoner et M. Hack ¹.

La Recherche en Allemagne et en France fut très active à la fin des années 70 alors que le groupe du MIT se dispersait. Progressivement des groupes de Recherche se formèrent sur le sujet en Europe et dans le Monde. Après un premier Symposium en 1975 au MIT qui n'eut pas de suite, il y eut des Conférences chaque année à partir de 1980. La première eu lieu symboliquement à Strasbourg, puis ensuite à Bad Honnef (RFA), Varenna (Italie), Toulouse, Aarhus (Danemark), Espoo (Finlande), Oxford, Zaragoza (Espagne), Venise, Bonn, Paris et Aarhus.

Les éditions Springer Verlag publient les résultats les plus importants sous l'intitulé *Advances in Petri Nets* dans la série *Lecture Notes in Computer Science*. Les numéros 254 et 255 correspondent en particulier à un cours donné à Bad Honnef. Les numéros les plus récents sont 266, 340 424, 483, 524 et 616.

Parmi les applications on peut citer: l'évaluation des performances des systèmes discrets, les protocoles de communication, la commande des ateliers de fabrication, la conception de logiciel temps-réel et/ou distribué, les systèmes d'information (organisation des entreprises), les interfaces homme-machine.

En ce qui concerne la conduite des Ateliers de Fabrication Automatisés, l'application des réseaux de Petri s'est effectuée d'abord en France sous la forme un peu altérée de la norme *Grafset* pour la programmation des Automates Programmables Industriels. Cette norme a

¹Il ne faut pas oublier la thèse de Vinton Gray Cerf "*Multiprocessors, semaphores and a graph model of computation*" (1972 University of California Los Angeles) qui à l'aide de *digraphs* a développé de façon indépendante des idées proches.

d'abord été imaginée par une commission de l'Afcet en 1977, puis elle est devenue une norme industrielle (C03.190-UTE) en 1980 en France, puis au niveau Européen par le bureau central de la CEI sous la référence IEC 848.

Si on se place à un niveau de coordination (coordination des activités de contrôleurs de machines et non Automates Programmables directement connectés aux capteurs et aux actionneurs), les pays les plus actifs sont l'Allemagne (Société PSI à Berlin) et le Japon (Société Hitachi). En France on peut citer les Sociétés IBSI Electronique, VERILOG avec une orientation plus "*Génie Logiciel*" ainsi que la société IXI qui a développé MISS-RdP, un outil de simulation permettant l'évaluation des performances en présence de pannes (réseaux de Petri temporels et stochastiques). Notons également l'outil "*Design/CPN*" (conception à l'aide des réseaux de Petri colorés) de la société Américaine Meta Software Corporation.

Chapitre 2

Définitions.

2.1 Concepts de base.

2.1.1 Réseau de Petri.

Un réseau de Petri est un quadruplet

$$\mathcal{R} = \langle P, T, Pre, Post \rangle \quad (2.1)$$

où:

- P est un ensemble fini de places,
- T est un ensemble fini de transitions,
- $Pre : P \times T \rightarrow N$ est l'application *places précédentes*,
- $Post : P \times T \rightarrow N$ est l'application *places suivantes*.

On utilise également la notation:

$$C = Post - Pre \quad (2.2)$$

et C est en général appelée *matrice d'incidence* du réseau de Petri.

2.1.2 Réseau marqué.

Un réseau marqué est le couple:

$$\mathcal{N} = \langle \mathcal{R}, M \rangle \quad (2.3)$$

où:

- \mathcal{R} est un réseau de Petri,
- M est le marquage initial; c'est une application

$$M : P \rightarrow N \quad (2.4)$$

$M(p)$ est le nombre de marques (jetons *tokens*) contenus dans la place p .

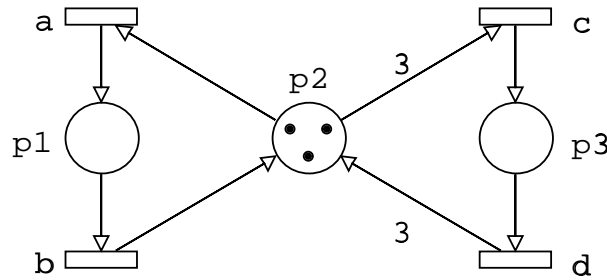


Figure 2.1: Exemple de réseau de Petri

2.1.3 Graphe associé et notations matricielle.

A un réseau de Petri on peut associer un graphe qui possède deux type de nœuds: les places et les transitions. Un arc relie une place p à une transition t si et seulement si $Pre(p, t) \neq 0$. Un arc relie une transition t à une place p si et seulement si $Post(p, t) \neq 0$. Les valeurs non nulles des matrices Pre et $Post$ sont associées aux arcs comme étiquettes (par défaut on prend la valeur 1). Un exemple de graphe associé à un réseau de Petri est donné par la figure 2.1.

Le marquage M peut être représenté par un vecteur ayant pour dimension le nombre de places; Pre , $Post$ et C seront alors des matrices dont le nombre de lignes est égal au nombre de places et le nombre de colonnes est égal au nombre de transitions.

On note $Pre(\cdot, t)$, $Post(\cdot, t)$ et $C(\cdot, t)$ les colonnes de ces matrices associées à une transition t . Ce sont des vecteurs ayant pour dimension le nombre de places (comme M).

Considérons par exemple la figure 2.1. Elle définit le réseau de Petri suivant:

- $P = \{p1, p2, p3\}$,
- $T = \{a, b, c, d\}$,

$$\bullet \quad Pre = \begin{array}{cccc} & a & b & c & d \\ \begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 3 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} & & & & \begin{array}{l} p1 \\ p2 \\ p3 \end{array} \end{array}$$

$$\bullet \quad Post = \begin{array}{cccc} & a & b & c & d \\ \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 3 \\ 0 & 0 & 1 & 0 \end{bmatrix} & & & & \begin{array}{l} p1 \\ p2 \\ p3 \end{array} \end{array}$$

$$\text{On a alors: } C = \begin{array}{cccc} & a & b & c & d \\ \begin{bmatrix} 1 & -1 & 0 & 0 \\ -1 & 1 & -3 & 3 \\ 0 & 0 & 1 & -1 \end{bmatrix} & & & & \begin{array}{l} p1 \\ p2 \\ p3 \end{array} \end{array}$$

$$\text{Et le marquage initial est: } M = \begin{array}{l} \begin{bmatrix} 0 \\ 3 \\ 0 \end{bmatrix} \\ p1 \\ p2 \\ p3 \end{array}$$

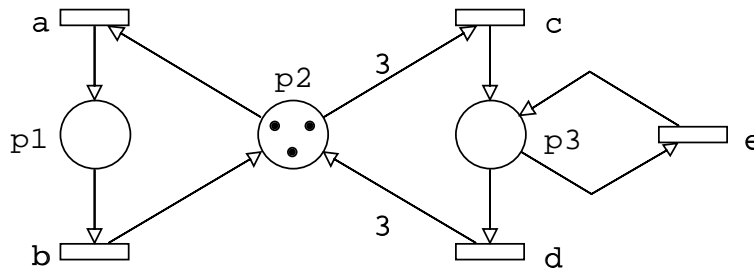


Figure 2.2: Exemple de réseau de Petri non pur

2.1.4 Réseau de Petri pur.

Un réseau de Petri R est pur si et seulement si:

$$\forall p \in P \text{ et } \forall t \in T : Pre(p, t).Post(p, t) = 0 \quad (2.5)$$

Le graphe ne comprend aucune boucle élémentaire, c'est-à-dire aucune transition n'ayant la même place en entrée et en sortie.

Par exemple le réseau de Petri de la figure 2.1 est pur. Par contre le réseau de la figure 2.2 ne l'est pas.

2.1.5 Transition franchissable.

Une transition t est franchissable (sensibilisée, *enabled*) si et seulement si:

$$\forall p \in P \quad M(p) \geq Pre(p, t) \quad (2.6)$$

On peut exprimer que t est franchissable par les notations:

$$M \geq Pre(., t) \quad (2.7)$$

$$M(t > \quad (2.8)$$

$$M \xrightarrow{t} \quad (2.9)$$

Par exemple dans le réseau de Petri de la figure 2.1, et pour le marquage initial M :

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

les transitions a et c sont franchissables car

$$Pre(., a) = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} \quad Pre(., c) = \begin{bmatrix} 0 \\ 3 \\ 0 \end{bmatrix}$$

et donc:

$$M > Pre(., a) \text{ et } M = Pre(., c)$$

2.1.6 Franchissement d'une transition.

Si t est franchissable pour le marquage M , le franchissement (tir, *firing*) de t donne le nouveau marquage M' tel que:

$$\forall p \in P \quad M'(p) = M(p) - Pre(p, t) + Post(p, t) \quad (2.10)$$

On utilise également les notations:

$$M' = M - Pre(., t) + Post(., t) \quad (2.11)$$

$$M(t > M') \quad (2.12)$$

$$M \xrightarrow{t} M' \quad (2.13)$$

Par exemple dans le réseau de la figure 2.1, après le franchissement de a à partir du marquage initial M , on obtient le marquage M' suivant:

$$\begin{bmatrix} 1 \\ 2 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 3 \\ 0 \end{bmatrix} - \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$$

2.1.7 Conflit et parallélisme.

Conflit structurel: Deux transitions $t1$ et $t2$ sont en conflit structurel si et seulement si elles ont au moins une place d'entrée en commun:

$$\exists p \quad Pre(p, t1) \cdot Pre(p, t2) \neq 0 \quad (2.14)$$

Conflit effectif: Elles sont en conflit effectif pour un marquage M si et seulement si $t1$ et $t2$ sont en conflit structurel et que:

$$M \geq Pre(., t1) \quad (2.15)$$

$$M \geq Pre(., t2) \quad (2.16)$$

Parallélisme structurel: Deux transitions $t1$ et $t2$ sont parallèles structurellement si:

$$(Pre(., t1))^T \times Pre(., t2) = 0 \quad (2.17)$$

Elles n'ont donc aucune place d'entrée commune (le produit scalaire de leurs vecteurs Pre est nul).

Parallélisme effectif: Deux transitions $t1$ et $t2$ sont parallèles pour un marquage donné M si et seulement si elles sont parallèles structurellement et:

$$M \geq Pre(., t1) \quad (2.18)$$

$$M \geq Pre(., t2) \quad (2.19)$$

Ainsi dans le réseau de Petri de la figure 2.1 les transitions a et c sont en conflit structurel puisque

$$Pre(p2, a) \cdot Pre(p2, c) = 3$$

par contre, les transitions b et d sont structurellement parallèles. En effet:

$$Pre(., b) = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \quad \text{et} \quad Pre(., d) = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

Pour le marquage initial M , les transitions a et c sont en conflit effectif. Si maintenant on considère le marquage:

$$M' = \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix}$$

alors les transitions b et d sont effectivement parallèles (elles peuvent être franchies indépendamment l'une de l'autre).

2.1.8 Séquence de franchissement.

Si $M_0 \xrightarrow{t_a} M_1$ et $M_1 \xrightarrow{t_b} M_2$ on dit que la séquence " $t_a; t_b$ " est franchissable à partir de M_0 , ce que l'on note:

$$M_0 \xrightarrow{t_a; t_b} M_2 \quad (2.20)$$

ou encore:

$$M_0(t_a; t_b) > M_2$$

Soit \bar{s} le vecteur dont les composantes $\bar{s}(t)$ sont les nombres d'occurrences des transitions t dans une séquence de franchissement s , ce vecteur est appelé *vecteur caractéristique* de s . Sa dimension est égale au nombre de transitions du réseau de Petri.

En considérant toujours le réseau de Petri de la figure 2.1, on a, par exemple pour la séquence " $s = a; a; b$ ":

$$\begin{bmatrix} 0 \\ 3 \\ 0 \end{bmatrix} \xrightarrow{a} \begin{bmatrix} 1 \\ 2 \\ 0 \end{bmatrix} \xrightarrow{a} \begin{bmatrix} 2 \\ 1 \\ 0 \end{bmatrix} \xrightarrow{b} \begin{bmatrix} 1 \\ 2 \\ 0 \end{bmatrix}$$

et

$$\overline{a; a; b} = \begin{bmatrix} 2 \\ 1 \\ 0 \\ 0 \end{bmatrix} \quad \begin{matrix} a \\ b \\ c \\ d \end{matrix}$$

Les évolutions du marquage d'un réseau de Petri sont alors données par l'équation:

$$M' = M - Pre.\bar{s} + Post.\bar{s} \quad (2.21)$$

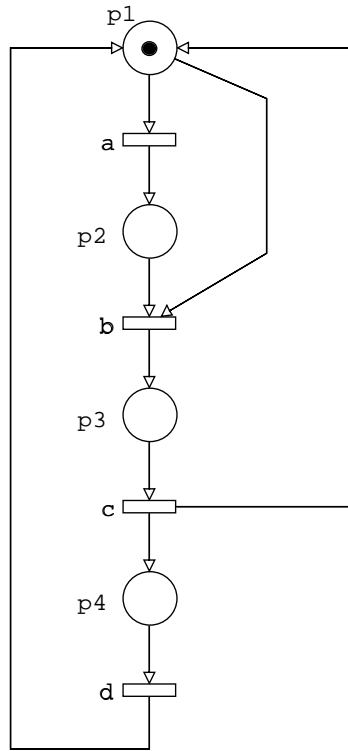


Figure 2.3: Exemple de séquence non franchissable

que l'on peut également écrire:

$$M' = M + C.\bar{s} \text{ avec } M \geq 0, \bar{s} \geq 0 \quad (2.22)$$

Cette équation est appelée l'équation fondamentale d'un réseau de Petri.

Attention: il n'est pas suffisant de trouver un vecteur caractéristique \bar{s} vérifiant l'équation 2.22 pour être sûr qu'il existe une séquence s effectivement franchissable du marquage M vers le marquage M' . En effet, il faut que le marquage de départ soit tel que les transitions seront effectivement franchissables pour chaque marquage intermédiaire. Considérons par exemple le réseau de Petri de la figure 2.3 et la séquence $s_{abcd} = a; b; c; d$. Si:

$$M = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad C = \begin{bmatrix} -1 & -1 & 1 & 1 \\ 1 & -1 & 0 & 0 \\ 0 & 1 & -1 & 0 \\ 0 & 0 & 1 & -1 \end{bmatrix} \quad \overline{s_{abcd}} = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}$$

alors:

$$M' = M + C.\overline{s_{abcd}},$$

mais en fait on peut vérifier que cette séquence " s_{abcd} " n'est pas franchissable et donc on ne peut pas écrire:

$$M \xrightarrow{a;b;c;d} M'.$$

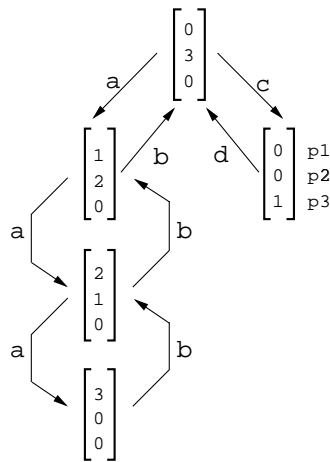


Figure 2.4: Exemple de graphe des marquages accessibles

2.1.9 Ensemble des marquages accessibles.

Soit un réseau de Petri $\mathcal{N} = \langle \mathcal{R}, M_0 \rangle$. L'ensemble des marquages accessibles $A(\mathcal{R}; M_0)$ d'un réseau de Petri marqué est l'ensemble des marquages que l'on peut atteindre à partir du marquage initial M_0 par une séquence de franchissement.

$$A(\mathcal{R}; M_0) = \{M_i, \exists s \ M_0 \xrightarrow{s} M_i\} \quad (2.23)$$

On peut, lorsque cet ensemble est fini, le représenter sous la forme d'un graphe $GA(\mathcal{R}; M_0)$. Ce graphe a pour ensemble de sommets l'ensemble des marquages accessibles $A(\mathcal{R}; M_0)$, un arc orienté relie deux sommets M_i et M_j s'il existe une transition t franchissable permettant de passer d'un marquage à un autre: $M_i \xrightarrow{t} M_j$.

En général, les arcs du graphe sont étiquetés par les transitions correspondantes. C'est en fait la machine à état *équivalente* au réseau de Petri. Il faut toutefois bien remarquer que la notion de processus ayant disparu on ne peut plus faire la distinction entre les transitions parallèles et les transitions en conflit pour un marquage donné.

La figure 2.4 représente le graphe des marquages accessibles pour le réseau de Petri de la figure 2.1 et le marquage initial M_0 .

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

2.2 Ensemble de règles et réseau de Petri.

2.2.1 Réseau de Petri.

Nous avons vu qu'un réseau de Petri pouvait être vu sous l'un des deux aspects suivants:

- Un graphe avec deux types de sommets et un comportement dynamique,

- Un ensemble de matrices d'entiers positifs ou nul dont le comportement dynamique est décrit par un système linéaire (équation fondamentale).

Il peut également être considéré comme un système de production de connaissance (système de déduction) encore appelé *système de règles* et fondé sur une représentation des connaissances sous la forme:

$$CONDITION \implies ACTION$$

2.2.2 Système de règles.

Un système de règles ou système de production de connaissance est formé:

- d'une *base de faits* (contexte des faits connus augmenté des règles permettant de déduire de nouveaux faits),
- d'un *mécanisme d'inférence* (c'est-à-dire de déduction).

Le mécanisme d'inférence le plus élémentaire consiste à ranger l'ensemble des règles dans une liste et à parcourir cette liste séquentiellement. Dès que la *condition* d'une règle est vraie dans le contexte courant (faits vérifiés au moment de l'inférence), la règle est appliquée. Si aucune règle n'est applicable ou si un fait terminal (conclusion) devient vrai, le mécanisme d'inférence s'arrête.

Le plus souvent, dans un contexte donné, plusieurs règles peuvent être applicables et le résultat de la déduction peut être différent suivant que l'on choisit d'appliquer d'abord telle ou telle règle ou de les appliquer toutes simultanément. Le mécanisme élémentaire choisit la première rencontrée sans même détecter que d'autres possibilités existaient.

Une situation où plusieurs règles sont applicables est appelée situation de *conflit*. La résolution des conflits, encore appelée le *contrôle*, caractérise en fait le mécanisme d'inférence qui peut alors devenir très complexe.

Dans le cas d'un réseau de Petri:

- les matrices *Pre* et *Post*, exploitées colonnes par colonnes, (c'est-à-dire l'ensemble des transitions avec leurs règles de franchissement) sont la base de règles,
- le marquage (initial) est le contexte (initial),
- la résolution des conflits dérive des notions de transitions parallèles (l'ordre de franchissement est indifférent) et de transitions en conflit (le résultat dépendra de l'ordre des franchissements); si deux transitions sont en conflit effectif, on en franchit une choisie de façon aléatoire et l'autre ne sera pas franchie.

2.2.3 Grammaire.

Un réseau de Petri est toutefois un système de règle bien particulier. La base de faits n'est pas générale car elle est construite uniquement à partir de la notion de marquage des places, c'est-à-dire de la valeur d'un ensemble fini de compteurs. Il est possible de décrire cela à

l'aide de mots construits sur un alphabet. L'alphabet est l'ensemble des identificateurs des places:

$$\{p1, p2, p3\},$$

par exemple.

Un marquage pour lequel la place $p1$ contient un jeton et la place $p3$ deux jetons s'écrira alors:

$$p1 p3 p3,$$

ou bien alors:

$$p1 p3^2.$$

On peut définir une application μ qui à tout marquage M et plus généralement à tout vecteur défini sur l'ensemble des places (les colonnes de Pre et de $Post$) associe un mot $\mu(M)$ de P^* . P^* est l'ensemble des suites finies d'éléments de P y compris l'élément vide λ .

La classe particulière de système de règles correspondant aux réseaux de Petri s'appelle une grammaire (système de réécriture de mots).

La grammaire $S = \langle P; Q \rangle$ associée au réseau $\mathcal{R} = \langle P, T; Pre, Post \rangle$ est définie par:

- son vocabulaire P ,
- l'ensemble Q des règles de réécriture:

$$t_i : \mu(Pre(\cdot, t_i)) \longrightarrow \mu(Post(\cdot, t_i)).$$

A un marquage initial M du réseau correspond un axiome (mot initial) $\mu(M)$ de la grammaire, à partir duquel on peut dériver de nouveaux mots.

Attention, quand on considère les séquences de franchissement si T est considéré comme un alphabet, T^* représente l'ensemble des séquences de franchissement et pour les éléments de T^* , l'ordre dans lequel les éléments apparaissent est important. Par contre les marquages sont des éléments de P^* pour lequel l'ordre n'a aucune importance.

2.2.4 Exemple.

Revenons au réseau de Petri de la figure 2.1. Si nous le considérons comme une grammaire nous aurons:

$$P = \{p1, p2, p3\}$$

et

$$Q = \begin{cases} a : p2 \rightarrow p1 \\ b : p1 \rightarrow p2 \\ c : p2^3 \rightarrow p3 \\ d : p3 \rightarrow p2^3 \end{cases}$$

Pour $\mu(M) = p2^3$, l'ensemble des marquages accessibles pourra alors s'écrire:

$$A(\mathcal{R}; M) = \{p2^3, p2^2 p1, p2 p1^2, p1^3, p3\}$$

Cet aspect est important car il permet des notations abrégées et surtout car il permet de faire le pont entre les réseaux de Petri et les techniques d'*Intelligence Artificielle*.

2.2.5 Conclusion.

Un réseau de Petri peut donc être vu de trois façons différentes, chacune présentant certains avantages et certains inconvénients.

Le graphe, lorsqu'il est de taille raisonnable, produit une description facilement transmissible permettant d'expliquer clairement certains mécanismes de synchronisation. Les jetons visualisent les objets et les circuits correspondent à des processus séquentiels répétitifs.

L'aspect matriciel, résumé par l'équation fondamentale 2.22, caractérise un sur-ensemble (car s doit être une séquence effectivement franchissable) de l'ensemble des marquages accessibles par un système d'équations linéaires. C'est pourquoi cet aspect est essentiel lors de l'analyse.

Enfin, nous venons de voir que l'on pouvait également considérer un réseau de Petri comme un système de règle particulier. Cet aspect est particulièrement intéressant quand on veut effectuer une mise en œuvre dans un contexte plus général orienté vers les techniques de l'Intelligence Artificielle.

2.3 “Bonnes” propriétés.

Nous allons définir un certain nombre de propriétés concernant les réseaux de Petri marqués et les réseaux de Petri non marqués.

Les premières (elles dépendent donc du marquage initial) sont en général regroupées sous le nom générique de “*bonnes*” propriétés. Leurs définitions impliquent des considérations sur l'ensemble des marquages accessibles à partir du marquage initial. Elles ne produisent donc pas directement des algorithmes permettant de déterminer si la propriété est vérifiée ou non car l'ensemble des marquages accessibles n'est pas toujours fini. Les méthodes d'analyse de ces propriétés seront données dans le chapitre “analyse des propriétés”.

Par contre, les propriétés concernant les réseaux de Petri indépendamment de leur marquage initial sont telles que des méthodes de calcul peuvent être dérivées directement des définitions par résolution d'un système d'équations linéaires. Ces propriétés seront définies dans le paragraphe suivant concernant les composantes conservatives et répétitives stationnaires.

2.3.1 K-borné.

a) Place k-bornée et binaire.

Une place p d'un réseau marqué N est k -bornée si et seulement si:

$$\forall M' \in A(\mathcal{R}; M_0) \quad M'(p) \leq k \quad (2.24)$$

Si $k = 1$ on dit que la place est *binaire* (*safe*).

Si on considère le réseau de Petri de la figure 2.1 on voit que pour le marquage initial $M_0 = p2^3$, la place $p3$ est binaire alors que les places $p1$ et $p2$ sont 3-bornées.

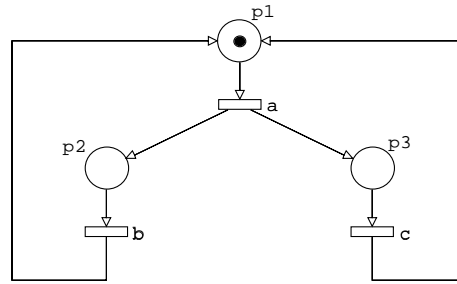


Figure 2.5: Réseau de Petri non borné

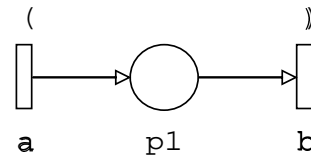


Figure 2.6: Réseau parenthèse

b) Réseau de Petri marqué k-borné et binaire.

Un réseau marqué N est k -borné (*bounded*) si et seulement si toutes ses places sont k -bornées.

Un réseau marqué N est *binaire (sauf) (safe)* si et seulement si toutes ses places sont binaires.

c) Exemples.

Le réseau de Petri de la figure 2.1 est 3-borné pour le marquage $M_0 = p2^3$. Si son marquage initial était $M_0 = p2$ il serait binaire (les transitions c et d ne seraient toutefois jamais franchies).

Considérons par contre le réseau de Petri de la figure 2.5 pour le marquage initial $M_0 = p1$. Chaque fois que l'on franchit la séquence $s = a; b$, on rajoute un jeton dans la place $p3$. Cette place est donc non bornée et le réseau l'est également (les places $p1$ et $p2$ sont également non bornées).

d) Remarques.

- Attention on considère un réseau marqué, c'est-à-dire un réseau de Petri avec un marquage initial donné.
- Le concept de réseau sauf présente l'intérêt suivant: si les places représentent des conditions logiques, la présence de plus d'un jeton dans une place signifie qu'une incohérence s'est glissée dans le modèle. Généralement il s'agit d'une condition logique mise à 1 lors d'un cycle de fonctionnement, qui n'a pas été utilisée (mise à 0) et qui est remise à 1 au cycle suivant.

- Le concept de réseau borné correspond au fait qu'un système physique est toujours limité. Toutefois, on peut être amené à utiliser des réseaux de Petri non bornés lorsque l'on veut évaluer les performances d'un système indépendamment des bornes de ses éléments de stockage intermédiaires.

Un exemple de réseau non borné décrivant un mécanisme classique est donné par la figure 2.6. Si au franchissement de la transition a on associe l'ouverture d'une parenthèse (parenthèse gauche) et à celui de la transition b la fermeture d'une parenthèse (parenthèse droite), les séquences de franchissement de transition qui à partir du marquage initial M_0 ($M_0(p1) = 0$) conduisent à ce même marquage M_0 décrivent toutes les expressions parenthésées licites. Il est clair que ce réseau n'est pas borné car on peut toujours ouvrir une parenthèse (franchir a) et donc rajouter un jeton dans la place $p1$.

2.3.2 Vivant.

a) Transition quasi-vivante.

Une transition t d'un réseau marqué \mathcal{N} avec

$$\mathcal{N} = \langle \mathcal{R}, M_0 \rangle$$

est quasi-vivante si et seulement si il existe une séquence de franchissement s telle que:

$$M_0 \xrightarrow{s} M' \quad \text{et} \quad M' \xrightarrow{t} \quad (2.25)$$

ce que l'on peut également écrire:

$$M_0 \xrightarrow{s;t}$$

b) Transition vivante.

Une transition t d'un réseau marqué \mathcal{N} avec

$$\mathcal{N} = \langle \mathcal{R}, M_0 \rangle$$

est vivante si et seulement si:

$$\forall M' \in A(\mathcal{R}; M_0) \quad \exists s \quad M' \xrightarrow{s;t} \quad (2.26)$$

c) Exemple.

Considérons le réseau de Petri de la figure 2.7. Le graphe des marquages associé au marquage initial $M_0 = p3 p4$ est donné par la figure 2.8. Il est clair que la transition d est quasi-vivante (elle peut être franchie une fois), mais non vivante (elle ne peut être franchie qu'une fois car on ne peut plus sortir de la composante fortement connexe pendante formée par les marquages $p1, p2, p3$ et $p4$).

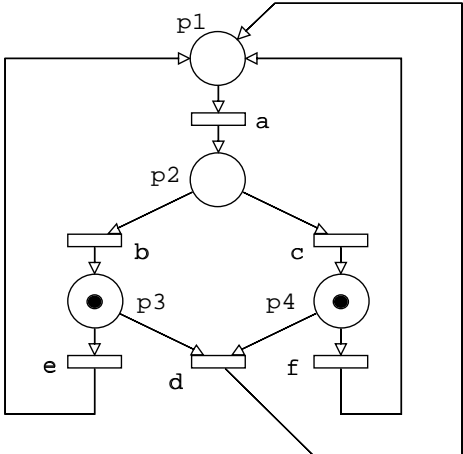


Figure 2.7: Transition quasi-vivante et non vivante

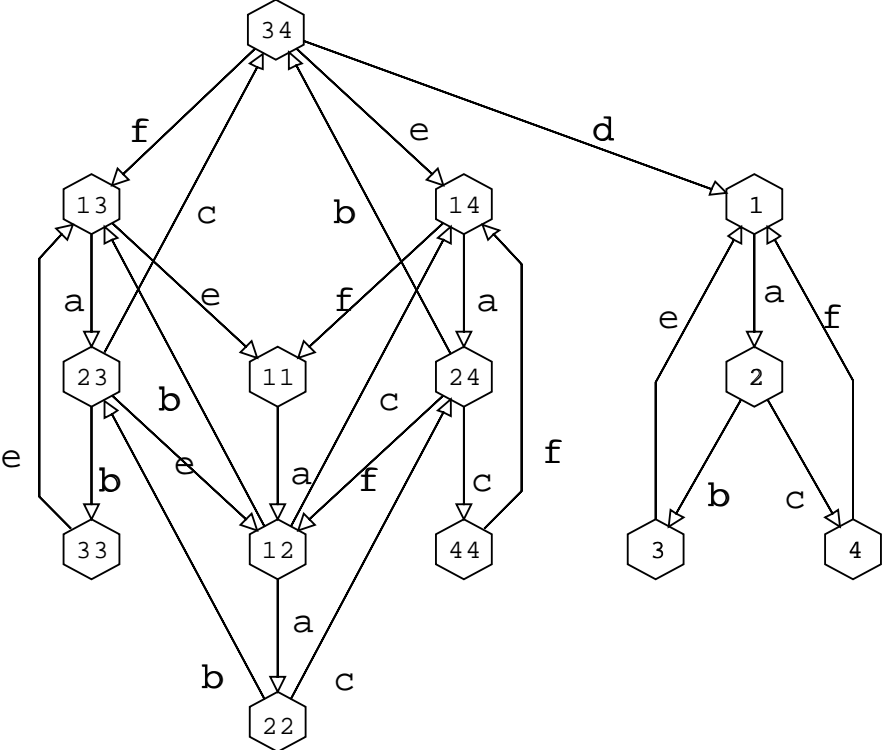


Figure 2.8: Graphe des marquages (transition quasi-vivante)

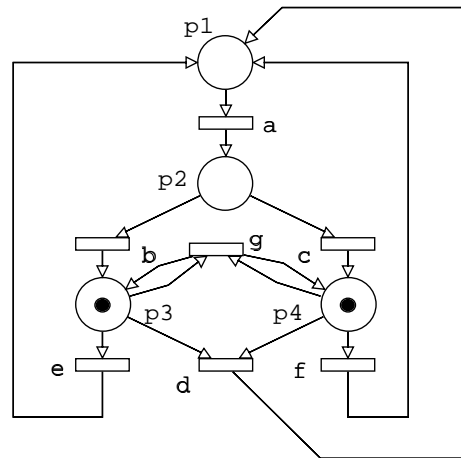


Figure 2.9: Transition quasi-vivante et séquence infinie

Par contre les transitions a , b , c , e et f sont vivantes car elles figurent dans la composante fortement connexe pendante et on pourra donc toujours trouver une séquence de franchissement les contenant.

Il faut faire attention qu'une transition peut, sans être vivante, apparaître une infinité de fois dans des séquences infinies de franchissement de transition. C'est le cas de la transition g dans la figure 2.9. Elle peut être franchie autant de fois que l'on veut avant le franchissement de d , mais après elle ne peut plus être franchie.

d) Réseau marqué vivant.

Un réseau de Petri marqué $\mathcal{N} = \langle \mathcal{R}, M_0 \rangle$ est vivant si et seulement si toutes ses transitions sont vivantes.

- Attention on considère un réseau de Petri pour un marquage initial donné (réseau marqué).
- Un réseau de Petri vivant garantit qu'aucun blocage ne peut être provoqué par la structure du réseau de Petri. Par contre, il ne prouve pas l'absence d'éventuels blocages provoqués par une mauvaise interaction entre le réseau de Petri et son environnement (voir plus loin les problèmes liés à l'interprétation d'un réseau de Petri).
- Un réseau de Petri vivant garantit également l'absence de partie morte (jamais atteinte).

Par exemple, le réseau de Petri de la figure 2.1 est vivant pour le marquage initial donné, par contre celui de la figure 2.7 ne l'est pas.

Un réseau de Petri peut très bien être non borné et vivant; c'est par exemple le cas du réseau de la figure 2.6 (la séquence $a b$ est toujours franchissable quel que soit le marquage).

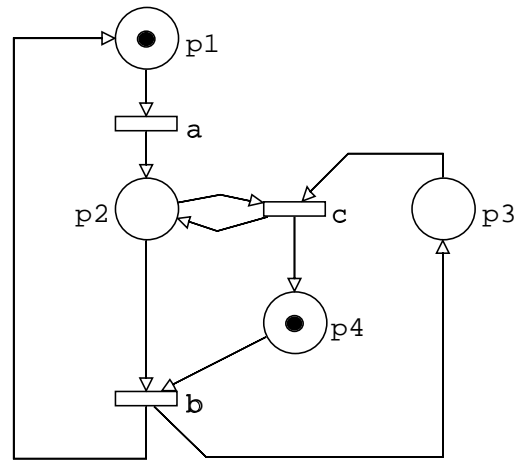


Figure 2.10: Exemple de réseau non réinitialisable

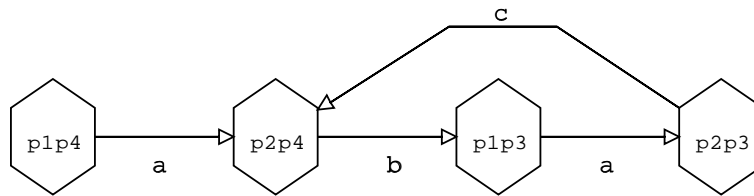


Figure 2.11: Graphe des marquages du réseau non réinitialisable

2.3.3 Réinitialisable.

a) Réseau marqué réinitialisable.

Un réseau marqué $\mathcal{N} = \langle \mathcal{R}, M_0 \rangle$ est réinitialisable (propre) si et seulement si son graphe des marquages accessibles $GA(\mathcal{R}; M_0)$ est fortement connexe:

$$\forall M' \in A(\mathcal{R}; M_0) \quad \exists s \quad M' \xrightarrow{s} M_0 \quad (2.27)$$

La plupart des systèmes ont des fonctionnements répétitifs et donc les réseaux de Petri utilisés seront réinitialisables.

b) Exemple.

Considérons le réseau marqué de la figure 2.10 dont le graphe des marquages accessibles est donné par la figure 2.11. Il est non réinitialisable puisqu'il n'existe aucune séquence permettant de revenir au marquage initial " $M_0 = p1 p4$ " après le franchissement de la transition a .

Par contre il est intéressant de noter qu'il est vivant car à l'intérieur de la composante fortement connexe pendante formée par les marquages $p2 p4$, $p1 p3$, $p2 p3$, il est toujours possible de franchir toutes les transitions du réseau de Petri soient a , b et c .

En conséquence, si on considère le réseau de la figure 2.10 pour l'un des marquages initiaux suivants: " $M_0 = p2 p4$ ", " $M_0 = p1 p3$ " ou " $M_0 = p2 p3$ ", il est à la fois vivant

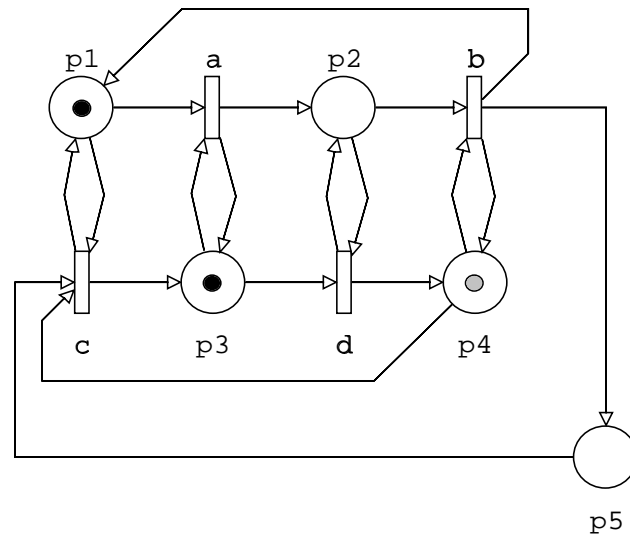


Figure 2.12: Exemple de propriétés dépendant du marquage initial

et réinitialisable. Inversement, le réseau de Petri de la figure 2.7 est réinitialisable *pour le marquage* “ $M_0 = p1$ ” sans être vivant puisque la transition d ne peut jamais être franchie.

2.3.4 Remarque finale.

Il est important de bien remarquer que les propriétés que nous venons de définir sont fortement liées au marquage initial. Considérons par exemple le réseau de Petri de la figure 2.12. Pour le marquage “ $M_0 = p1 p3$ ” il est binaire, vivant et réinitialisable. Si on rajoute un jeton dans la place $p4$ alors il cesse d’être borné (voir la séquence $abab\dots$). Enfin pour le marquage “ $M_0 = p1$ ” le réseau est binaire et non vivant (aucune transition franchissable).

2.4 Composantes conservatives et répétitives stationnaires.

2.4.1 Composantes conservatives, invariants de place.

Nous allons maintenant chercher des propriétés découlant directement de la *structure* du réseau de Petri, c’est-à-dire ne dépendant pas de son marquage initial.

Considérons le réseau de Petri de la Figure 2.13, et regardons le sous-réseau formé par les places 1 et 2 et par les transitions a et b . Considérons maintenant la somme:

$$M(1) + M(2)$$

qui vaut 1 pour le marquage initial M_0 . Le franchissement de a ne change rien à cette somme, ni celui de b . Celui des autres transitions du réseau ne modifient pas non plus cette somme. On peut donc affirmer que pour tous les marquages accessibles à partir du marquage initial on a:

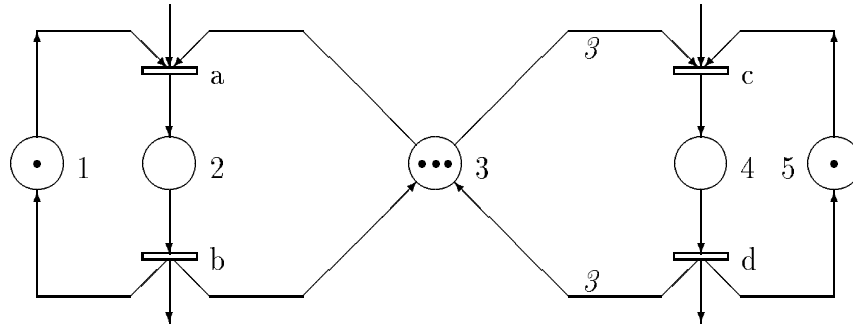


Figure 2.13: Invariants

$$M(1) + M(2) = 1$$

En fait, quel que soit le marquage initial M_0 nous aurons:

$$M(1) + M(2) = M_0(1) + M_0(2) \quad \forall M \in A(\mathcal{R}; M_0)$$

C'est pourquoi le sous-réseau formé de l'ensemble des places 1 et 2 et des transitions qui y sont connectées est appelé *composante conservative*. La forme linéaire $M(1) + M(2)$ étant appelée *invariant linéaire de place*.

Si on considère maintenant le sous-réseau formé par les places 2, 3 et 4 et par les transitions qui y sont connectées (a , b , c et d), on pourra vérifier que

$$M(2) + M(3) + 3.M(4) = 3 \quad \forall M \text{ accessible}$$

Un *invariant linéaire* de place est une fonction linéaire du marquage des places dont la valeur est une constante, ne dépendant que du marquage initial du réseau. Il correspond à une contrainte sur les états et les activités du système qui sera toujours vérifiée, quelles que soient ses évolutions.

Si nous revenons à l'équation fondamentale (équation 2.22), en prémultipliant chacun des termes par un vecteur f^T (dans un premier temps on ne fait pas de restriction sur f c'est-à-dire que l'on suppose que ses composantes sont des réels):

$$f^T.M' = f^T.M + f^T.C.\bar{s} \quad (2.28)$$

il est clair que la seule façon de se rendre indépendant des séquences de franchissement s est d'annuler $f^T.C$.

Une *composante conservative* d'un réseau de Petri est une solution (f étant un vecteur de réels) de l'équation:

$$f^T.C = 0 \quad (2.29)$$

Une *composante conservative*, du point de vue graphique, définit un *sous-réseau* de Petri. On ne considère que les places pour lesquelles la composante de f est non nulle (c'est le support du vecteur) et les transitions d'entrée et de sortie de ces places. Les transitions comportent le même nombre d'arcs en entrée et en sortie (dans les cas simples).

Si f est solution de l'équation ci-dessus alors la fonction linéaire:

$$f^T.M = f^T.M_0 \quad \forall M \in A(\mathcal{R}; M_0) \quad (2.30)$$

est l'*invariant linéaire* de place correspondant.

Cette équation dépend du marquage initial, alors que la composante conservative en était complètement indépendante.

Composantes conservatives positives:

Certains auteurs se restreignent aux solutions f positives et résolvent donc:

$$f^T.C = 0 \quad \text{avec } f > 0 \quad (2.31)$$

car ce sont souvent les plus intéressantes comme nous le verrons par la suite. De plus, comme C est une matrice d'entiers, les solutions de l'équation seront des vecteurs de rationnels et il sera toujours possible de se ramener à des solutions de vecteurs d'entiers.

2.4.2 Composantes répétitives, invariants de transition.

Considérons à nouveau la figure 2.13, le franchissement de la séquence $c d$ à partir du marquage initial redonne le même marquage. Le sous-réseau à observer est cette fois formé à partir des transitions c et d et en prenant toutes les places qui sont connectées à ces transitions. Ce sous-réseau est une *composante répétitive stationnaire* car chaque place de ce sous-réseau n'a qu'un arc entrant et un arc sortant dans le sous-réseau. On peut faire le même raisonnement avec la séquence $a b$, par exemple.

Un *invariant de transition* est une séquence de franchissements de transitions qui ne modifie pas le marquage du réseau. Un tel invariant correspond à une séquence cyclique d'événements qui peut être répétée indéfiniment. Si on reprend l'équation fondamentale (2.22) il est clair que tout invariant de transition doit être une séquence s telle que \bar{s} vérifie: $C.\bar{s} = 0$.

Toute solution \bar{s} de l'équation:

$$C.\bar{s} = 0. \quad (2.32)$$

est appelée *composante répétitive stationnaire*.

On considère d'abord que \bar{s} est un vecteur de réels, mais comme précédemment les solutions seront des rationnels.

Une *composante répétitive stationnaire* définit un *sous-réseau* si l'on ne considère que les transitions pour lesquelles la composante correspondante de \bar{s} est non nulle avec leur places d'entrée et de sortie.

Si \bar{s} est le vecteur caractéristique d'une séquence s de franchissements de transitions effectivement franchissable à partir d'un marquage accessible alors cette séquence s est un *invariant de transitions*.

La composante répétitive stationnaire ne dépend pas du marquage initial, par contre l'existence ou non d'un invariant de transition correspondant en dépend fortement.

Composantes répétitives stationnaires positives:

Il est clair que si l'une des composantes de \bar{s} est négative on ne trouvera aucune séquence de franchissement de transition correspondante (il s'agirait d'un franchissement vers l'arrière). C'est pourquoi on se restreint généralement aux solutions pour lesquelles les composantes de \bar{s} sont des entiers et telles que:

$$C.\bar{s} = 0 \text{ avec } \bar{s} > 0 \tag{2.33}$$

Chapitre 3

Analyse des Propriétés.

On va maintenant donner des *algorithmes* permettant de déterminer si un réseau de Petri donné possède les bonnes propriétés. En fait, bien que les propriétés de vivant et réinitialisable aient été montrées décidables, aucun algorithme praticable n'a été trouvé dans le cas général. Nous nous contenterons de donner les principes de la preuve de décidabilité concernant la propriété de borné, puis nous donnerons une méthode pour déterminer les autres propriétés. Cette méthode n'est effectivement applicable que pour des réseaux possédant toutes les "bonnes" propriétés. Nous présenterons ensuite d'autres méthodes permettant de s'affranchir partiellement de l'énumération des marquages. Ensuite nous donnerons une méthode systématique pour le calcul des composantes conservatives et répétitives stationnaires et enfin nous décrirons les relations existant entre les "bonnes" propriétés (qui dépendent du marquage initial) et les invariants.

3.1 Analyse par énumération des marquages.

3.1.1 La propriété de réseau k-borné est décidable.

Il faut d'abord remarquer qu'il est équivalent de dire que le réseau marqué est borné et que le nombre de ses marquages accessibles est fini puisque le nombre de places est fini. D'autre part, il est équivalent de dire que l'ensemble des marquages accessibles n'est pas fini et qu'il existe au moins une séquence infinie de franchissement qui ne soit pas répétitive stationnaire (c'est-à-dire qu'elle n'est pas construite à partir d'une séquence finie que l'on répète indéfiniment et qui passe toujours par les mêmes marquages).

Si on se contente d'énumérer les marquages accessibles, dans le cas où leur nombre est fini il y aura arrêt de la procédure, par contre si le nombre de marquages est infini la procédure ne s'arrêtera jamais. Il n'est donc pas évident a priori que la propriété de k-borné soit décidable.

La décidabilité découle des propriétés suivantes:

1. Monotonie:

$$\exists M, M', s \text{ tels que } M \xrightarrow{s} \text{ et } (M' > M) \implies (M' \xrightarrow{s})$$

Quand une évolution est possible à partir d'un marquage donné, elle reste *possible* si on augmente le marquage (ajouter des jetons dans certaines places).

2. Si $\exists M, M' \in A(\mathcal{R}; M)$ tels que $M \xrightarrow{s} M'$ et $M' > M$ alors le réseau de Petri marqué n'est pas k-borné. En effet, on déduit immédiatement du point précédent que la séquence s peut être répétée autant de fois qu'on le désire, et à chaque fois, on accroît le contenu en jeton d'au moins une place puisque M' est strictement plus grand que M .
3. (Lemme de Karp et Miller) Toute suite infinie de vecteurs formés d'entiers positifs ou nuls

$$v_1, v_2, \dots, v_k, \dots$$

est telle qu'elle contient au moins deux éléments (en fait une infinité de couples) v_i et v_j avec $i < j$ telles que $v_i < v_j$ ou bien $v_i = v_j$.

Le résultat est évident pour les vecteurs n'ayant qu'une composante: il est impossible d'avoir une suite infinie d'entiers positifs ou nuls qui soit strictement décroissante.

4. Considérons maintenant une séquence infinie s de franchissements de transitions, la suite des marquages est une suite infinie de vecteurs d'entiers positifs ou nuls. Le lemme précédent implique donc l'existence d'une sous-séquence s_1 telle que

$$M \xrightarrow{s_1} M'$$

et

$$M \leq M'$$

et que l'on peut répéter indéfiniment.

Les séquences définies au point "2" sont donc caractéristiques des séquences de longueur infinies non stationnaires (qui ne repassent jamais par les mêmes marquages). Elles sont donc caractéristiques du fait qu'un réseau n'est pas borné, c'est-à-dire que l'ensemble des marquages accessibles $A(\mathcal{R}; M)$ contient une infinité d'éléments.

3.1.2 Arbre de couverture.

L'algorithme permettant de décider si un réseau de Petri marqué est k-borné est fondé sur la construction d'un arbre appelé *arbre de couverture*. On part du marquage initial. Chaque transition franchissable pour ce marquage donne naissance à une branche, on calcule les marquages obtenus par le franchissement des transitions et pour chaque marquage obtenu on recommence. On arrête la construction d'une branche lorsque:

1. on trouve un marquage égal à un marquage déjà rencontré et pour lequel tous les successeurs ont déjà été calculés ou bien vont être calculés car ils correspondent à des branches pendantes qui seront explorées ensuite (sinon on réexplorerait un sous-arbre déjà calculé),
2. on trouve un marquage strictement supérieur à un marquage *de la branche en cours d'exploration*.

Dans le deuxième cas, on arrête l'exploration de l'arbre car le réseau marqué n'est pas borné (voir paragraphe précédent). Sinon on continue jusqu'à ce que toutes les branches aient été explorées. Alors on est certain que le réseau marqué est borné et l'on connaît l'ensemble (et le graphe) des marquages accessibles. L'algorithme se termine (donc c'est effectivement un algorithme car tout algorithme doit se terminer après un nombre fini de pas) car:

- on ne peut avoir aucune branche de longueur infinie (lemme de Karp et Miller),
- le nombre de branches est fini car pour chaque marquage le nombre de transitions franchissables est fini (inférieur au nombre de transitions du réseau de Petri).

3.1.3 Recherche des propriétés.

Nous venons de voir que la propriété de "borné" était décidable pour un réseau marqué. La construction de l'arbre de couverture donne un algorithme pour répondre à cette question.

Pour les autres propriétés, il n'existe pas d'algorithme général bien qu'elles soient, d'un point de vue théorique, décidables. Toutefois, la plupart du temps, on désire que le réseau construit possède *simultanément* toutes les "bonnes" propriétés. La démarche est alors la suivante: on montre que le réseau est k-borné, puis qu'il est réinitialisable puis enfin qu'il est vivant.

En effet on a les propriétés suivantes:

$$\mathcal{R} \text{ réinitialisable pour } M \iff GA(\mathcal{R}, M) \text{ fortement connexe}$$

ainsi que:

$$\mathcal{R} \text{ réinitialisable pour } M \implies (\mathcal{R} \text{ quasi-vivant} \iff \mathcal{R} \text{ vivant}).$$

Pour prouver que la propriété "réseau \mathcal{R} réinitialisable pour M " implique que $GA(\mathcal{R}, M)$ est fortement connexe, il suffit de passer par le marquage initial M pour montrer qu'il existe au moins un chemin entre deux marquages accessibles quelconques M_i et M_j .

Pour prouver que quasi-vivant implique vivant si le réseau est réinitialisable, il suffit à partir d'un marquage accessible M_i de revenir au marquage initial M pour trouver une séquence amenant au franchissement d'une transition quelconque.

Finalement nous avons donc la démarche suivante:

1. on prouve que le réseau marqué est borné par la construction de l'arbre de couverture,
2. si le réseau marqué est borné, on construit le graphe des marquages accessibles,
3. on montre que le graphe des marquages accessibles est fortement connexe,
4. on vérifie que toute transition apparaît au moins une fois comme étiquette d'un arc du graphe des marquages accessibles.

Cette démarche prouve que le réseau marqué est à la fois borné, réinitialisable et vivant.

3.2 Analyse par réduction.

La méthode présentée au paragraphe précédent est extrêmement lourde à mettre en œuvre lorsque la taille de l'ensemble des marquages accessibles devient grande (explosion combinatoire des états). L'idée est alors de trouver des règles de réduction qui soient telles que le réseau de Petri de départ et le réseau réduit aient les mêmes propriétés (équivalence vis-à-vis des "bonnes propriétés") de façon à n'appliquer la méthode décrite ci-dessus qu'à des réseaux de taille raisonnable.

Attention cette notion d'équivalence n'implique pas nécessairement que les deux réseaux ont les mêmes séquences de tir de transitions ou les mêmes invariants de place ou de transition. D'autre part, il faut remarquer que dans certains cas particuliers certaines règles ne conservent pas toutes les bonnes propriétés.

3.2.1 Place substituable.

Informellement, une place substituable est une place qui sert de simple relai entre deux transitions (ou deux ensembles de transitions). Le franchissement de la transition d'entrée (de l'une des transitions d'entrée) est une condition suffisante pour pouvoir franchir la transition de sortie (une des transitions de sortie, respectivement). La simplification consiste à fusionner les transitions d'entrée et les transitions de sortie deux à deux.

Plaçons nous d'abord dans le cas simple où la place substituable p n'a qu'une transition en entrée et une en sortie.

Soit t_s la transition de sortie de p ($Pre(p, t_s) \neq 0$), soit t_e la transition d'entrée de p ($Post(p, t_e) \neq 0$), on doit avoir :

- l'égalité des poids des arcs entrant et sortant de p

$$Post(p, t_e) = Pre(p, t_s) \quad (3.1)$$

- pas d'autre place en entrée de t_s :

$$\forall p' \in P \text{ si } p' \neq p \text{ alors } Pre(p', t_s) = 0 \quad (3.2)$$

Le réseau réduit est obtenu en remplaçant les transitions t_e et t_s par une seule transition t_{es} et en effaçant p avec :

$$\begin{aligned} \forall p' \in P \quad Pre(p', t_{es}) &= Pre(p', t_e) \\ Post(p', t_{es}) &= Post(p', t_e) + Post(p', t_s) \end{aligned} \quad (3.3)$$

Il est important de souligner qu'une longue séquence linéaire va se compacter en une seule transition par application répétitive de cette règle de réduction.

Si la place substituable p possède plusieurs transitions en entrée et en sortie, alors tous les poids des arcs doivent être égaux et il faut considérer tous les couples formés par une

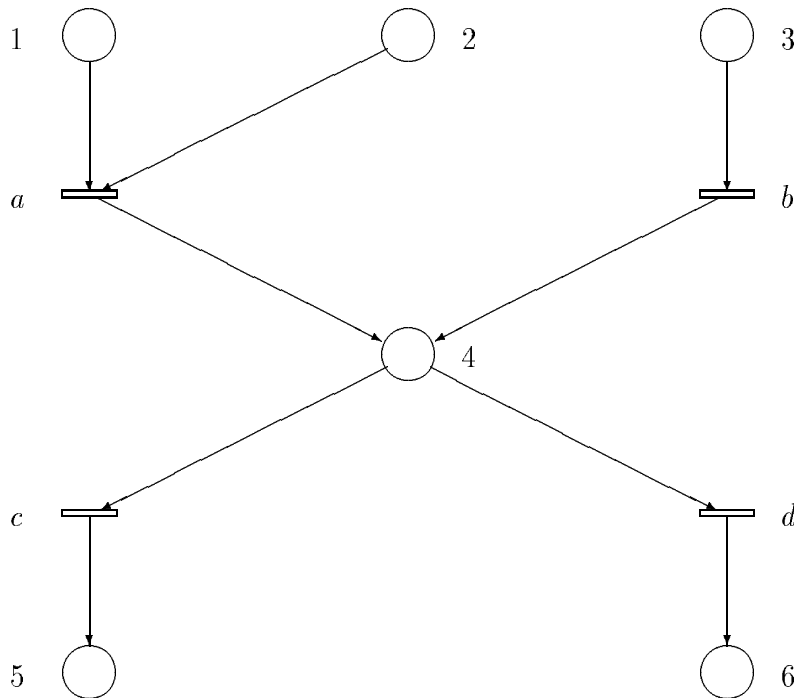


Figure 3.1: Place substituable

transition d'entrée et une transition de sortie. Considérons par exemple le réseau de Petri de la Figure 3.1 et vérifions que la place 4 est substituable. On doit avoir

$$Post(4, a) = Post(4, b) = Pre(4, c) = Pre(4, d)$$

Le franchissement de l'une des deux transitions a ou b est une condition suffisante pour pouvoir franchir soit c soit d . La place 4 ne sert que de relais et on peut la supprimer. Sa suppression produit le réseau de Petri de la Figure 3.2 avec les quatre transitions ac , ad , bc et bd .

Attention si la place contient initialement des jetons, il faut franchir la transition de sortie avant de supprimer la place substituable. Mais dans ce cas *il n'y a plus équivalence pour la propriété de réinitialisable*.

Attention, si la transition de sortie de la place substituable ne possède pas de place de sortie, *il n'y a plus équivalence vis-à-vis de la propriété de k -borné*.

3.2.2 Place implicite.

a) Principe:

Une place implicite est “une place qui ne sert à rien”. En effet, son marquage est une combinaison linéaire du marquage d'un ensemble de places E et, vis-à-vis de ses transitions de sortie elle n'introduit aucune condition supplémentaire de franchissement (en général par rapport aux places de E).

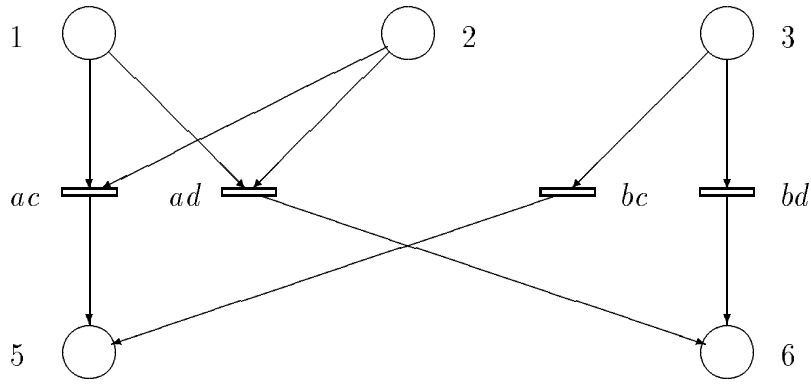


Figure 3.2: Simplification d'une place substituable

Le fait que le marquage d'une place implicite doit être une combinaison linéaire du marquage d'un ensemble E relie cette notion à celle de composante conservative.

b) Exemple:

Considérons le réseau de la Figure 3.3, nous allons montrer que la place 1 est implicite vis-à-vis de l'ensemble des places $\{2, 3\}$. Pour cela faisons comme si nous voulions chercher les composantes conservatives.

Le jeu d'équations produit par $f^T \cdot C = 0$ est:

$$\begin{aligned} f_1 + f_2 - f_4 &= 0 \\ -f_2 + f_3 &= 0 \end{aligned} \tag{3.4}$$

$$-f_1 - f_3 + f_5 = 0 \tag{3.5}$$

Comme on ne s'intéresse qu'aux places "1", "2" et "3" on cherche d'abord des solutions pour lesquelles f_4 et f_5 sont égaux à zéro. On trouve alors:

$$f_1 = -f_2$$

et

$$f_2 = f_3$$

et on obtient donc l'invariant linéaire de place:

$$M(1) = M(2) + M(3)$$

correspondant à la composante conservative:

$$\begin{bmatrix} -1 \\ 1 \\ 1 \\ 0 \\ 0 \end{bmatrix} .$$

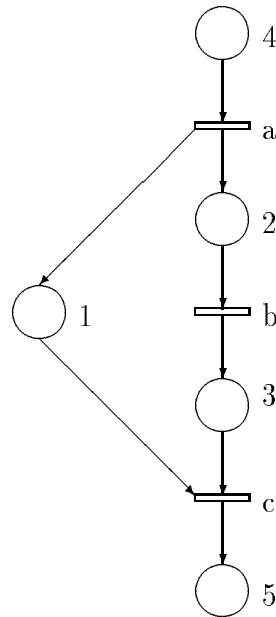


Figure 3.3: Place implicite

Nous pouvons donc à tout instant calculer le marquage de la place 1 en fonction du marquage des places 2 et 3.

La seule transition de sortie de la place 1 est la transition c . Or pour qu'elle soit franchissable, il faut que la place 3 contienne au moins un jeton et nous avons vu plus haut que cela implique que 1 contient également au moins un jeton. Donc la place 1 *ne sert à rien* et peut être supprimée sans modification des séquences de franchissement des transitions. Cela donne alors le réseau réduit de la Figure 3.4.

c) Remarques:

Il faut faire attention à deux choses. D'abord, le marquage initial du réseau modifie l'équation permettant de calculer le marquage de la place implicite. Par exemple, la présence d'un jeton initialement dans l'une des places 2 ou 3 donnera:

$$M(1) = M(2) + M(3) - 1.$$

En conséquence le fait que la place 2 contienne un jeton n'est pas suffisant pour affirmer que 1 contient au moins un jeton.

Le second problème auquel il faut faire attention est qu'il n'est pas suffisant que le marquage d'une place soit fonction du marquage d'un sous-ensemble E pour qu'elle soit implicite. Un contre exemple est donné par la Figure 3.5, la transition d n'étant pas transition suivante des places 2 ni 3, la place 1 ne peut être supprimée. En effet, dans ce réseau, la

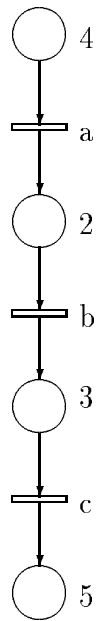


Figure 3.4: Simplification de la place implicite

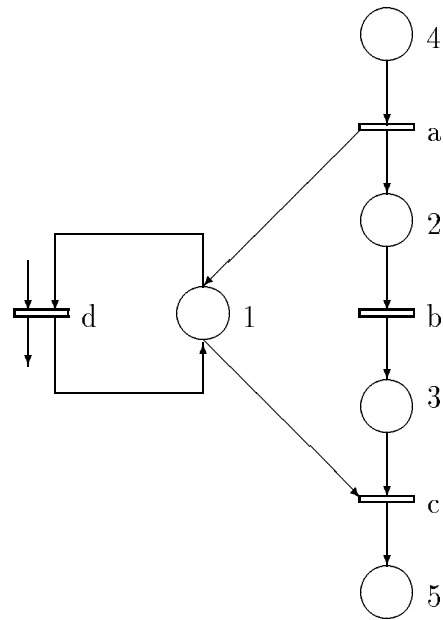


Figure 3.5: Contre exemple de place implicite

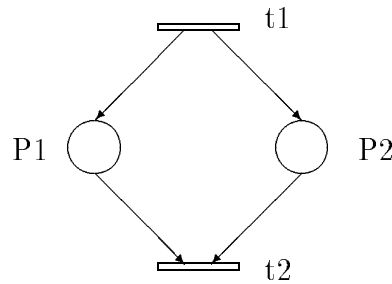


Figure 3.6: Place identiques

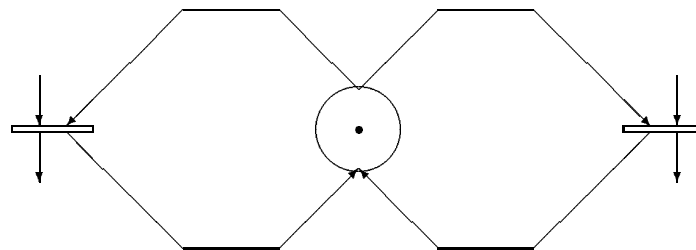


Figure 3.7: Place implicite dégénérée

transition d ne peut être franchie qu'entre le tir de a et celui de c . Cette condition de franchissement serait enlevée si la place 1 était supprimée.

d) Places identiques et places implicites dégénérées:

Deux types de places implicites sont évidents:

1. Si deux places $p1$ et $p2$ sont identiques (mêmes transitions en entrée et en sortie avec les mêmes poids sur les arcs), alors $p1$ est implicite vis-à-vis de $p2$,
2. Si une place n'est connectée au réseau de Petri que par des boucles élémentaires, elle est implicite vis-à-vis de l'ensemble vide (son marquage est constant). On l'appelle place implicite dégénérée.

Ceci est illustré par les Figures 3.6 et 3.7.

3.2.3 Transitions neutres ou identités.

Une transition neutre est "une transition qui ne sert à rien". En effet, elle n'est connectée au réseau de Petri que par des boucles élémentaires. Son franchissement ne modifie donc pas le marquage du réseau de Petri.

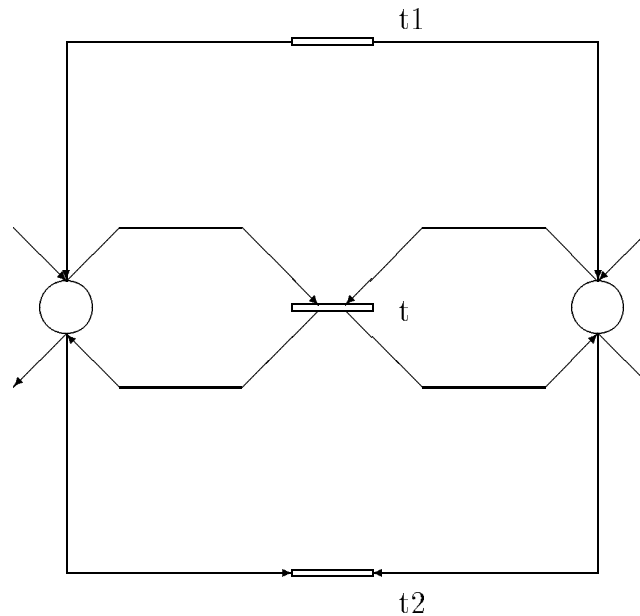


Figure 3.8: Transition neutre

Une transition t est *neutre* (on dit aussi que t est une transition *identité*) si et seulement si:

$$Pre(., t) = Post(., t) \quad (3.6)$$

Pour pouvoir supprimer une transition neutre t en conservant la propriété de réseau marqué vivant, il faut être sûr que t est vivante sinon c'est une partie morte du réseau qui sera supprimée lors de la réduction. Pour cela il faut qu'il existe une autre transition qui, si elle est vivante dans le réseau réduit, garantit que t est vivante dans le réseau initial. Il existe deux cas évidents de telle transition: soit elle produit un marquage sensibilisant t , soit elle possède les mêmes conditions de franchissement que t . Pour supprimer une transition neutre t il faut donc qu'il existe soit $t1$ tel que:

$$Post(., t1) = Pre(., t) \quad (3.7)$$

soit $t2$ tel que:

$$Pre(., t2) = Pre(., t) \quad (3.8)$$

Ceci est illustré par la figure 3.8. Un contre exemple est donné par le réseau de Petri de la figure 3.9. En effet, la transition d , bien que ne transformant pas le marquage et étant franchissable à partir du marquage initial, ne peut être éliminée. On obtiendrait un réseau vivant (voir figure 2.10) alors que d n'est jamais plus franchissable après le franchissement de a . On peut en effet vérifier que le réseau de la figure 3.9 est non réinitialisable (son graphe des marquages accessibles est identique à celui de la figure 2.11).

Par contre dans le réseau de Petri de la figure 2.9 g est neutre et peut être éliminée car la transition d possède les mêmes condition de franchissement qu'elle.

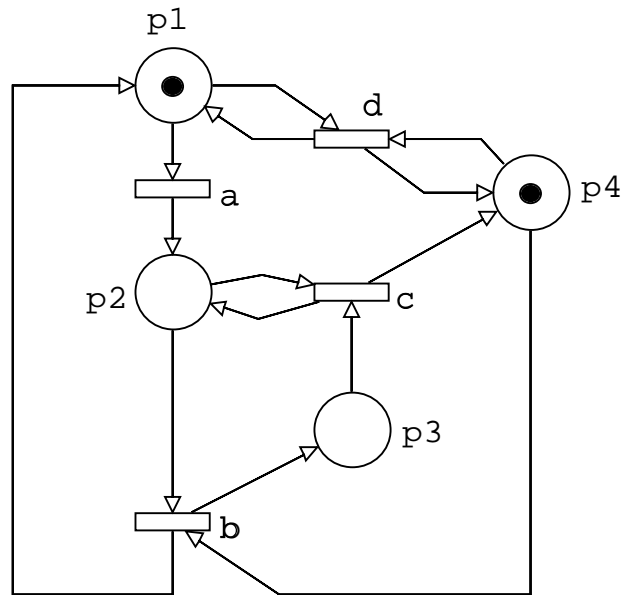


Figure 3.9: Transition neutre non simplifiable

3.2.4 Transitions identiques.

Deux transitions sont identiques si les colonnes correspondantes des matrices Pre et $Post$ sont identiques. Il est clair que l'une des deux peut être supprimée sans changer les propriétés du réseau de Petri.

Les transitions $t1$ et $t2$ sont *identiques* si et seulement si:

$$Pre(\cdot, t1) = Pre(\cdot, t2) \quad (3.9)$$

$$Post(\cdot, t1) = Post(\cdot, t2) \quad (3.10)$$

Le réseau de Petri de la Figure 3.10 donne un exemple de deux transitions $t1$ et $t2$ qui sont identiques. La simplification vis-à-vis de la transition $t2$ donne le réseau de Petri de la Figure 3.11.

3.3 Calcul des composantes conservatives et répétitives stationnaires.

3.3.1 Principe du calcul d'une base.

Nous expliquerons la méthode sur les composantes conservatives, il suffit en effet de transposer la matrice C (intervertir places et transitions) pour trouver les composantes répétitives stationnaires.

On cherche donc les solutions de l'équation:

$$f^T.C = 0$$

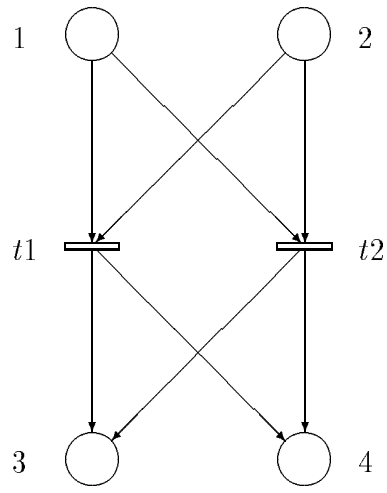


Figure 3.10: Exemple de transitions identiques

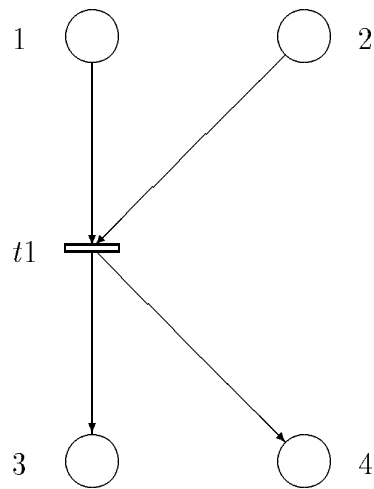


Figure 3.11: Simplification de transitions identiques

qui s'écrit également:

$$C^T \cdot f = 0$$

Elle comporte autant de variables que de places (les n composantes de f ou les n lignes de C) et autant d'équations que de transitions (les m colonnes de C).

Cette équation matricielle est équivalente au système d'équations:

$$\left. \begin{array}{l} c_{11} \cdot f_1 + \dots + c_{n1} \cdot f_n = 0 \\ \dots + \dots + \dots = 0 \\ c_{1m} \cdot f_1 + \dots + c_{nm} \cdot f_n = 0 \end{array} \right\} m \text{ équations}$$

Il s'agit d'un système d'équations linéaires dont le second membre est nul. L'ensemble des solutions forme donc un espace vectoriel. Il existe toujours la solution dégénérée $f^T = 0$ mais elle ne présente aucun intérêt. Ce qui est intéressant, c'est de chercher une base de l'espace des solutions.

Comme nous le verrons par la suite, si r est le rang de C et n le nombre de places, la dimension de cet espace vectoriel des solutions sera:

$$\dim p = n - r.$$

Celui des composantes répétitives stationnaires étant

$$\dim t = m - r$$

m étant le nombre de transitions.

Il existe une méthode systématique de résolution des équations linéaires, il s'agit de la *méthode de Gauss*. Cette méthode est basée sur le fait que l'on ne modifie pas les solutions d'un système d'équations linéaire en effectuant les opérations suivantes sur les lignes (ou les colonnes):

1. échange de deux lignes,
2. multiplication d'une ligne par un scalaire non nul,
3. addition d'une ligne à une autre ligne.

Ici, comme nous cherchons à obtenir une base des solutions, nous allons travailler sur les lignes de C c'est-à-dire que nous allons effectuer des changements de variables de façon à chercher à triangulariser cette matrice.

Nous effectuerons des combinaisons des changements de variables suivants:

1. on remplace la variable f_i par $a \cdot f'_i$ avec $a \neq 0$,
2. on remplace la variable f_i par $f'_i + f'_j$ (ce qui revient comme nous le verrons par la suite à remplacer la ligne i par la somme des lignes i et j sans toucher à la ligne j),
3. on réordonne les variables quand c'est nécessaire.

Ici C n'est pas carrée, on triangularisera une sous-matrice S de dimension r correspondant à un système d'équations linéairement indépendantes qui ne possèdent donc qu'une solution (dans notre cas la solution dégénérée).

Les colonnes n'appartenant pas à S (sous-matrice S') seront alors des combinaisons linéaires des colonnes de S , et les $n - r$ lignes de C n'appartenant pas à S ne contiendront que des zéros. En effet, elles doivent correspondre à des variables n'apparaissant dans aucune équation et dont la valeur reste libre. Si $n - r = 0$ cela veut dire que seule la solution dégénérée existe.

On se ramène donc au système d'équations suivant:

$$f'^T.C' = 0 \quad \text{avec} \quad \begin{array}{l} f = F.f' \\ \text{et} \quad C' = F^T.C \end{array}$$

où F est une matrice régulière décrivant le changement de variables et où la matrice C' possède la forme suivante:

$$C' = \left[\begin{array}{c|c} S & S' \\ \hline (0) & (0) \end{array} \right]$$

avec:

$$S = \left[\begin{array}{cccccc} s_{11} = 1 & s_{12} & s_{13} & \dots & s_{1r} \\ 0 & s_{22} = 1 & s_{23} & \dots & s_{2r} \\ 0 & 0 & s_{33} = 1 & \dots & s_{3r} \\ \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & \dots & s_{rr} = 1 \end{array} \right]$$

S' étant une matrice quelconque.

Si on écrit les r premières équations de ce nouveau système, on obtient:

$$\begin{array}{rcl} f'_1 & = & 0 \\ s_{12}.f'_1 + f'_2 & = & 0 \\ \dots & = & 0 \\ s_{1r}.f'_1 + s_{2r}.f'_2 + \dots + f'_r & = & 0 \end{array}$$

Ce système possède une solution unique vis-à-vis des r variables f'_1, \dots, f'_r qui est la solution dégénérée. Donc les solutions, sous la forme f' , sont telles que les r premières composantes de ce vecteur sont toujours égales à zéro. Par contre, les $dimp = n - r$ autres composantes peuvent être choisies librement puisqu'elles ont été éliminées du système d'équations (les $n - r$ dernières lignes de C' sont nulles).

Sous cette forme, l'obtention d'une base de l'espace vectoriel des solutions est donc immédiate. Il suffit de choisir des solutions pour lesquelles une et une seule des composantes

de f' est non nulle:

$$\left[\begin{array}{c} (0) \\ \hline f'_{r+1} = 1 \\ 0 \\ \dots \\ 0 \end{array} \right] \left[\begin{array}{c} (0) \\ \hline 0 \\ f'_{r+2} = 1 \\ \dots \\ 0 \end{array} \right] \dots \left[\begin{array}{c} (0) \\ \hline 0 \\ 0 \\ \dots \\ f'_n = 1 \end{array} \right]$$

d'où la dimension $n - r$ de l'espace vectoriel des solutions.

Pour pouvoir écrire cette base sous la forme initiale de l'équation

$$f^T.C = 0$$

il suffit donc de savoir construire la matrice F . On obtient ainsi une base de composantes conservatives.

Comme nous l'avons dit, la méthode de triangularisation de Gauss consiste à faire apparaître des zéros en combinant les lignes de la matrice. Supposons que les $i - 1$ premières lignes et colonnes aient été triangularisées. On considère la ligne i (une variable) et la colonne i (une équation) et on annule les coefficients de cette colonne correspondant aux variables $i + 1$ jusqu'à n en multipliant la ligne i par c_{ki} , la ligne k par c_{ii} et en retranchant la ligne i de la ligne k .

Si le coefficient c_{ii} est nul, la démarche est impossible. On effectue alors des permutations parmi les $m - i$ dernières colonnes de C (en cours de transformation) pour faire apparaître un c_{ii} non nul. Si cela est impossible, cela veut dire que la ligne i ne comprend que des zéros puisque les $i - 1$ premières colonnes de C ont déjà été traitées. La variable f_i a alors déjà été éliminée du système d'équations, elle produira un élément de la base de l'espace vectoriel des solutions. On permute alors les $n - i$ dernières lignes de C pour faire apparaître une ligne ne contenant pas que des zéros. Si cela est impossible, c'est que l'algorithme est terminé; C a été ré-écrite sous la forme C' .

Remplacer la "ligne k " par " c_{ii} fois la ligne k moins c_{ki} fois la ligne i " revient à faire le changement de variable suivant (il est licite pour $c_{ii} \neq 0$):

$$\begin{aligned} f_i &= f'_i - c_{ki}.f'_k \\ f_k &= c_{ii}.f'_k \\ f_j &= f'_j \quad \forall j \neq i, k \end{aligned}$$

En effet, l'équation j (colonne j de C), par exemple, qui s'écrivait:

$$c_{1j}.f_1 + \dots + c_{ij}.f_i + \dots + c_{kj}.f_k + \dots + c_{nj}.f_n = 0$$

devient:

$$c_{1j}.f'_1 + \dots + c_{ij}.(f'_i - c_{ki}.f'_k) + \dots + c_{kj}.c_{ii}.f'_k + \dots + c_{nj}.f'_n = 0$$

soit:

$$c_{1j}.f'_1 + \dots + c_{ij}.f'_i + \dots + (c_{ii}.c_{kj} - c_{ki}.c_{ij}).f'_k + \dots + c_{nj}.f'_n = 0$$

Si on considère l'équation i ($j = i$) on vérifie d'autre part que la variable f'_k est ainsi éliminée de l'équation. La normalisation de la ligne i dans l'équation i (pour que le coefficient s_{ii} se S soit égal à 1) implique normalement de faire également le changement:

$$f_i = \frac{1}{c_{ii}} \cdot f'_i$$

mais en fait cette normalisation est inutile car elle ne modifie en rien le principe de la méthode et risque d'introduire des nombres fractionnaires dans les calculs.

La matrice F peut également être construite par transformations successives. On part d'une matrice identité et on effectue *sur les colonnes* de F les mêmes opérations que sur les lignes de C . Dans le cas précédent cela revient à remplacer la colonne " k " par " c_{ii} fois la colonne k moins c_{ki} fois la colonne i " dans la matrice F . Cette démarche est justifiée par le fait que ce changement de variable est décrit par:

$$f = \begin{matrix} 1 \\ i \\ k \\ n \end{matrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & -c_{ki} & 0 \\ 0 & 0 & c_{ii} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot f'$$

et que la matrice F du changement de variables résultant de l'enchaînement de changements de variables élémentaires A_1, A_2, \dots, A_l est égale à:

$$F = A_1 \cdot A_2 \cdot \dots \cdot A_l$$

3.3.2 Exemple.

Considérons par exemple le réseau de Petri de la figure 2.13. Au départ nous avons:

$$C_0 = \begin{bmatrix} -1 & 1 & 0 & 0 \\ 1 & -1 & 0 & 0 \\ -1 & 1 & -3 & 3 \\ 0 & 0 & 1 & -1 \\ 0 & 0 & -1 & 1 \end{bmatrix} \quad F_0 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

L'élimination de la ligne 1 (elle ne sera plus touchée et correspondra à la première ligne de S ; elle ne fera pas partie de la base engendrant l'espace vectoriel des solutions) vis-à-vis de la première colonne se fait en multipliant la première ligne ($C(1, \cdot)$) par -1 , en substituant $C(2, \cdot)$ par $C(2, \cdot) + C(1, \cdot)$ et $C(3, \cdot)$ par $C(3, \cdot) - C(1, \cdot)$. Ce qui produit:

$$C_1 = \begin{bmatrix} 1 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & -3 & 3 \\ 0 & 0 & 1 & -1 \\ 0 & 0 & -1 & 1 \end{bmatrix} \quad F_1 = \begin{bmatrix} -1 & 1 & -1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

S'il est impossible de faire apparaître un 1 dans une ligne de zéros, pour continuer on peut, par exemple, interchanger les lignes 2 et 5 ainsi que les colonnes b et d . Cela donne (les

réordonnements des colonnes, c'est-à-dire de l'ordre d'écriture des équations, n'altèrent pas F):

$$C_2 = \begin{bmatrix} 1 & 0 & 0 & -1 \\ 0 & 1 & -1 & 0 \\ 0 & 3 & -3 & 0 \\ 0 & -1 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad F_2 = \begin{bmatrix} -1 & 0 & -1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 \end{bmatrix}$$

On peut alors éliminer la ligne 2 de l'espace des solutions en substituant

- $C_2(3, \cdot)$ par $C_2(3, \cdot) - 3.C_2(2, \cdot)$
- et $C_2(4, \cdot)$ par $C_2(4, \cdot) + C_2(2, \cdot)$.

On obtient alors:

$$C_3 = \begin{bmatrix} 1 & 0 & 0 & -1 \\ 0 & 1 & -1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad F_3 = \begin{bmatrix} -1 & 0 & -1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & -3 & 1 & 0 \end{bmatrix}$$

Les trois dernières lignes étant nulles, la triangularisation est terminée. L'espace vectoriel des solutions est de dimension 3 et les trois dernières colonnes de F_3 sont une base de cet espace (elles correspondent aux trois dernières composantes des solutions en f' c'est-à-dire après changement de variables).

D'où les composantes conservatives:

$$\begin{bmatrix} -1 \\ 0 \\ 1 \\ 0 \\ -3 \end{bmatrix} \quad \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 1 \end{bmatrix} \quad \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

Ce qui donne les invariants linéaires suivants pour le marquage initial indiqué sur la figure 2.13:

$$\begin{aligned} -M(1) + M(3) - 3M(5) &= -1 \\ M(1) + M(2) &= 1 \\ M(4) + M(5) &= 1 \end{aligned}$$

3.3.3 Algorithme simplifié et recherche des solutions positives.

En fait seul le calcul des colonnes de la matrice F correspondant aux $n - r$ composantes de la base des solutions f est utile. Les autres colonnes de F peuvent donc être effacées.

A chaque pas (élimination d'une variable), la colonne de C (l'équation qui va être mise dans la base des équations) qui a servi à éliminer une variable (tous les éléments de la colonne

sont nuls sauf un) est effacée ainsi que la ligne correspondant à cette variable. La variable ne faisant pas partie de la base des solutions, la colonne correspondante de F est également effacée.

Enfin au lieu de construire F explicitement, on mémorise les opérations effectuées sur les lignes de C sous la forme de sommes formelles. C'est-à-dire que chaque colonne de F est mémorisée sous la forme d'une combinaison linéaire des colonnes d'origine ($f_i + f_j$) décrit une colonne obtenue en faisant la somme des colonnes i et j de F_0 .

D'autre part les solutions positives sont en général les plus intéressantes (c'est évident pour les composantes répétitives stationnaires). Il est donc utile de favoriser les combinaisons de ligne positives lors de l'élimination.

D'où l'algorithme simplifié suivant:

1. On cherche une colonne n'ayant qu'un seul terme différent de zéro. Elle correspond à une équation de la base des équations, cette colonne est effacée et la variable correspondant au terme non nul est éliminée puisque la seule solution possible étant zéro, elle ne peut faire partie de la base des solutions. On efface donc la ligne de C associée à cette variable. En résumé on efface une ligne et une colonne de C et la colonne correspondante de F . L'étape **1** est faite tant que possible, puis on va en **2**.
2. On cherche une colonne n'ayant qu'une seule composante non nulle d'un signe donné, toutes les autres étant soit nulles, soit du signe contraire (une positive et toutes les autres négatives ou nulles par exemple). La variable correspondant à cette ligne est alors éliminée en n'effectuant que des combinaisons positives de lignes pour faire apparaître les zéros dans la colonne considérée. Après cette opération, on efface cette colonne, la ligne de C correspondant à la variable éliminée et la colonne de F associée. Si une telle colonne a été trouvée on retourne en **1** sinon on va en **3**.
3. On cherche une colonne ayant $i \geq 2$ composantes non nulles positives et $j \geq 2$ composantes non nulles négatives. A l'aide de l'une des composantes non nulles positives on annule $j - 1$ composantes négatives. On retourne alors à l'étape **2** pour éliminer la variable correspondant à la seule composante négative non annulée. Il faut remarquer que l'on a ainsi choisi une composante positive parmi i et une composante négative parmi j . Il y a donc $i \times j$ façons de procéder. Ce choix peut avoir des conséquences sur la forme de la base obtenue. Si on ne trouve aucune colonne répondant au critère ci-dessus, on poursuit en **4**.
4. Si on trouve une colonne ayant toutes ses composantes non nulles avec le même signe, on poursuit l'élimination de Gauss pour cette colonne, mais la base ne comprendra pas que des solutions positives. Ensuite on va en **1**. Sinon c'est que l'algorithme est terminé. Soit toutes les colonnes sont nulles, soit la matrice ne comprend plus aucun élément. Les combinaisons non effacées correspondant à la construction de colonnes de F donnent directement les vecteurs de la base.

Reprenons l'exemple du réseau de la figure 2.13. On démarre avec:

$$C_0 = \begin{bmatrix} -1 & 1 & 0 & 0 \\ 1 & -1 & 0 & 0 \\ -1 & 1 & -3 & 3 \\ 0 & 0 & 1 & -1 \\ 0 & 0 & -1 & 1 \end{bmatrix} \begin{bmatrix} f_1 \\ f_2 \\ f_3 \\ f_4 \\ f_5 \end{bmatrix}$$

On considère la première colonne de C_0 , l'équation correspondante est donc sélectionnée pour faire partie de la base des équations. On va éliminer la variable f_2 . Pour cela il suffit d'ajouter la ligne correspondant à f_2 aux lignes correspondant à f_1 et f_3 . Avec ces nouvelles variables, l'équation ne comportera qu'un terme non nul. On efface alors la ligne f_2 (f_2 sera hors base de solutions) ainsi que la première colonne de C_0 . On obtient alors:

$$C_1 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & -3 & 3 \\ 0 & 1 & -1 \\ 0 & -1 & 1 \end{bmatrix} \begin{bmatrix} f_1 + f_2 \\ f_2 + f_3 \\ f_4 \\ f_5 \end{bmatrix}$$

On utilise la deuxième colonne de C_1 pour mettre f_4 hors base des solutions. Pour cela on ajoute trois fois la ligne " f_4 " à la ligne " $f_2 + f_3$ " et une fois à " f_5 ". On efface alors la deuxième colonne de C_1 ainsi que la ligne correspondant à f_4 . Cela donne:

$$C_2 = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} f_1 + f_2 \\ f_2 + f_3 + 3f_4 \\ f_4 + f_5 \end{bmatrix}$$

Il ne reste que des zéros. L'algorithme est terminé. Il produit la base correspondant aux sommes figurant à la droite de la matrice et décrivant les manipulations de ligne de C et de colonnes de F effectuées. En remplaçant les symboles f_i par la colonne i de F_0 , on obtient la base, soit:

$$\begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad \begin{bmatrix} 0 \\ 1 \\ 1 \\ 3 \\ 0 \end{bmatrix} \quad \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 1 \end{bmatrix}$$

3.4 Relation entre les diverses méthodes d'analyse.

3.4.1 Obtention de "bonnes" propriétés par le calcul des composantes.

La méthode d'analyse fondée sur l'énumération des marquages accessibles ne permet pas de montrer qu'un réseau marqué non borné est vivant ou réinitialisable. L'application de la règle "place substituable" permet dans certains cas cette analyse si on arrive à transformer un réseau non borné en réseau borné.

L'utilisation des invariants de place permet de montrer que certaines places sont bornées et de calculer leur borne sans énumérer l'ensemble des marquages accessibles. Par contre les invariants de transition ne donnent que des conditions nécessaires mais non suffisantes.

On dit qu'il existe une *couverture* de composantes conservatives (ou respectivement répétitives stationnaires) si l'on trouve une composante *positive* ou un ensemble de composantes élémentaires *positives* qui passe par toutes les places (respectivement transitions).

On a alors les deux résultats suivants.

Couverture de place et borné

Un réseau de Petri pour lequel existe une couverture de composantes conservatives $f^T > 0$ est *k-borné quel que soit son marquage initial*.

La forme linéaire $f^T.M = f^T.M_0$ permet de calculer une borne pour chaque place. En effet, puisque les composantes de f et les marquages sont des nombres non négatifs, nous avons pour une place p quelconque:

$$f(p).M(p) \leq f^T.M_0$$

et en conséquence la borne suivante (qui n'est pas nécessairement atteinte):

$$M(p) \leq \frac{f^T.M_0}{f(p)} \quad (3.11)$$

Il faut toutefois souligner qu'un réseau peut être borné pour un marquage donné sans posséder une couverture de composantes conservatives. Le réseau de Petri de la figure 2.12 en est un exemple pour le marquage initial $M = p1 p3$.

Ce résultat peut être expliqué simplement en considérant la figure 3.12. Les résultats obtenus à partir d'une couverture sont vrais pour l'ensemble des marquages "3" qui recouvre l'ensemble des marquages accessibles ("1") mais qui ne se confond pas nécessairement avec lui.

Borné, vivant et couverture de transition

Tout réseau de Petri qui est à la fois vivant et borné pour au moins un marquage initial est tel qu'une couverture de composantes répétitives stationnaires $\bar{s} > 0$ existe.

Ceci découle du fait que:

- borné \implies nombre fini de marquages sensibilisant une transition donnée,
- vivant \implies séquences de longueur infinie,

D'où nous avons:

borné + vivant \implies séquence répétitive stationnaire.

Il est clair que la réciproque de ce résultat est fautive, il suffit en général de choisir un marquage pour lequel toutes les places sont vides.

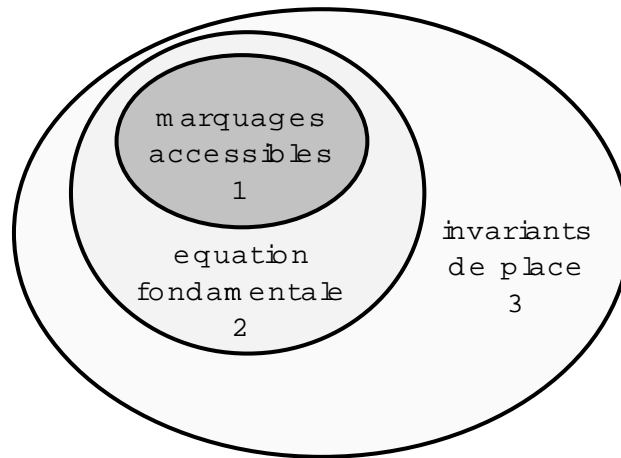


Figure 3.12: Caractérisation des marquages

3.4.2 Complémentarité entre l'analyse par réduction et le calcul des composantes.

Quand on trouve, après réduction, un réseau de Petri qui ne possède pas les *bonnes* propriétés pour un marquage initial donné, il est parfois difficile de le corriger. En effet, l'analyse par réduction, permet d'éviter l'énumération des marquages, mais ne produit pas, en cas d'anomalie, une séquence de franchissements de transitions menant à un marquage posant un problème (blocage mortel ou marquage supérieur à un marquage déjà atteint).

Le calcul des composantes conservatives et répétitives stationnaires fournit alors des informations complémentaires qui peuvent être très utiles. En effet d'après le paragraphe précédent les places non bornées seront telles qu'aucune composante conservative *positive* ne passe par elles. D'autre part, dans la plupart des cas, les transitions non vivantes seront telles qu'aucune composante répétitive stationnaire *positive* ne passe par elles.

3.4.3 Caractérisation des marquages.

Nous avons vu, tout au long de ce chapitre, plusieurs façons de caractériser un ensemble de marquages. La première façon consiste à énumérer les marquages effectivement *accessibles* à partir d'un marquage initial donné, en construisant des séquences de franchissements licites de transitions (ne franchir que des transitions sensibilisées). La seconde façon de caractériser un ensemble de marquages est de considérer l'équation caractéristique (2.22) et de calculer tous les marquages M' qui sont solutions de cette équation avec un vecteur $\bar{s} > 0$. On obtient un ensemble qui inclut nécessairement le premier mais qui peut être plus grand que lui. En effet, on peut trouver des vecteur $\bar{s} > 0$ ne correspondant à aucune séquence s franchissable. Enfin, l'ensemble des composantes conservatives $\{f_i\}$ associé à un marquage initial M_0 définit également un ensemble de marquages. Comme les f_i forment un espace vectoriel, il suffit de prendre une base B et de considérer l'ensemble des M' tels que:

$$f_{iB}^T \cdot M' = f_{iB}^T \cdot M_0 \quad \forall f_{iB} \in B$$

Cet ensemble inclut le précédent mais peut être strictement plus grand que lui.

La figure 3.12 illustre ce phénomène. L'ensemble "1" correspond aux marquages accessibles $A(\mathcal{R}, M_0)$, l'ensemble "2" aux marquages solutions de l'équation fondamentale

$$\{M' \mid M' = M_0 + C.\bar{s} \quad \text{avec} \quad M' \geq 0 \quad \text{et} \quad \bar{s} \geq 0\}$$

et l'ensemble "3" celui qui est défini par l'ensemble des invariants linéaires de place.

Illustrons ceci sur un exemple. Soit le réseau de la figure 3.13. L'ensemble des marquages accessibles se réduit à $\{p1, p2\}$.

L'ensemble des vecteurs $\bar{s} > 0$ produisant un marquage positif est formé des vecteurs :

$$\bar{s}_1 = [1 \ 0 \ 0 \ 0] \quad \bar{s}_2 = [1 \ 1 \ 1 \ 0] \quad \bar{s}_3 = [1 \ 1 \ 1 \ 1]$$

et, puisque la transformation de marquage correspondant au vecteur \bar{s}_3 est nulle, de tous les vecteurs :

$$\begin{aligned} \bar{s} &= \bar{s}_1 + \lambda.\bar{s}_3 \\ \bar{s} &= \bar{s}_2 + \lambda.\bar{s}_3 \\ \bar{s} &= \lambda.\bar{s}_3 \end{aligned}$$

où λ est un entier positif ou nul.

Ces vecteurs correspondent aux séquences "a", "a;b;c", "(a;b;c;d)*", "(a;b;c;d)*;a" etc. Seules les séquences "a;b" et "(a;b;c;d)*;a;b" sont interdites. La séquence "a" étant la seule effectivement possible, les autres passent par un marquage intermédiaire telle que $M(p_1) = -1$ ce qui est interdit.

Les transformations de marquage correspondant aux vecteurs \bar{s} sont:

$$\begin{bmatrix} -1 \\ 1 \\ 0 \\ 0 \end{bmatrix} \quad \begin{bmatrix} -1 \\ 0 \\ 0 \\ 1 \end{bmatrix} \quad \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

L'ensemble des marquages M' , solutions de l'équation fondamentale est alors: $\{p1, p2, , p4\}$.

Enfin l'ensemble des composantes conservatives f est décrit par la base suivante:

$$\begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}$$

L'ensemble "3" de la figure 3.12 est alors l'ensemble des M' vérifiant l'équation:

$$M'(p1) + M'(p2) + M'(p3) + M'(p4) = 1$$

Et l'ensemble des marquages vérifiant cette équation est: $\{p1, p2, p3, p4\}$.

Remarque: Dans de très nombreux cas pratiques les ensembles "1", "2" et "3" de la figure 3.12 sont identiques.

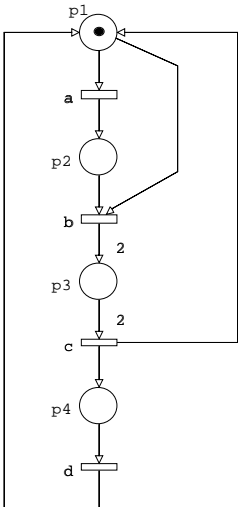


Figure 3.13: Réseau avec ensembles de marquages différents

Chapitre 4

Les données et le temps.

4.1 Introduction.

Dans ce chapitre, nous allons voir comment le réseau de Petri peut être utilisé pour décrire des enchaînements de traitements qui portent sur des structures de données complexes, communiquent avec un environnement et sont soumis à des contraintes temporelles explicites.

Nous verrons d'abord comment un réseau de Petri peut être interprété pour modéliser de tels systèmes.

Ensuite nous verrons comment on peut obtenir une description compacte et structurée en utilisant des réseaux de Petri *de haut niveau*, c'est-à-dire en associant plus étroitement certaines des données au réseau de Petri. Ces réseaux de Petri *de haut niveau* sont les réseaux de Petri Colorés, les réseaux Prédicats-Transitions et les réseaux à Objets.

Enfin, nous verrons diverses façons d'expliciter le temps dans les réseaux de Petri sans avoir recours à des variables booléennes dépendant du temps (monostables). Nous introduisons ainsi: les réseaux de Petri temporisés, temporels et stochastiques.

4.2 Interprétation d'un réseau de Petri.

4.2.1 Sémantique des places, des transitions et des jetons.

Interpréter un réseau de Petri, c'est d'abord donner un sens concret à un modèle mathématique en associant les places, les transitions et les jetons à des entités existantes.

Ainsi, les places peuvent être interprétées comme des activités d'un système à événements discrets, comme des stocks d'un atelier de production flexible, comme des conditions logiques d'un système logique séquentiel, comme des procédures en cours d'exécution dans un système informatique.

Aux transitions on peut faire correspondre des événements supposés de durée nulle à l'échelle de temps considérée. On peut également leur associer des activités ou des procédures à condition qu'elles soient *indivisibles* ou encore *ininterruptibles*. De telles transitions sont alors des *abréviations* pour des séquences élémentaires formées d'une transition de début, d'une place décrivant l'activité ou la procédure et d'une transition de fin (voir figure 4.1).

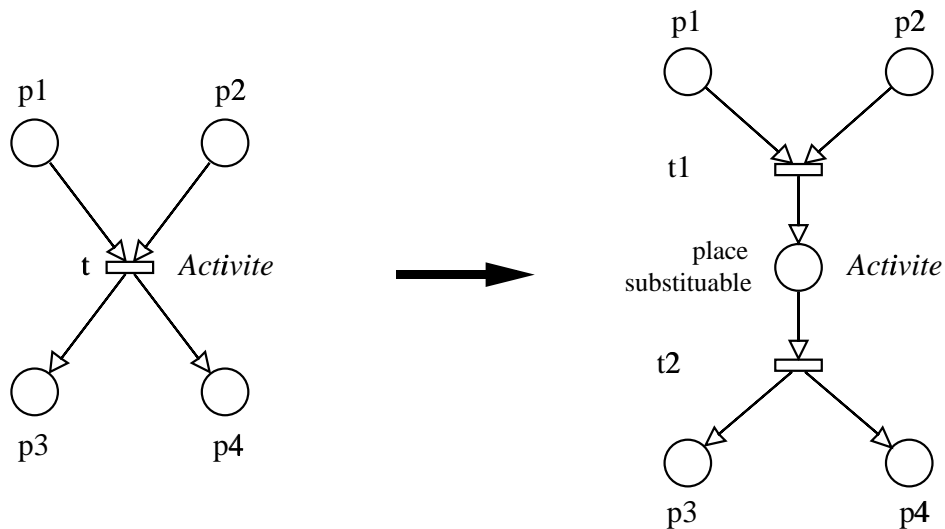


Figure 4.1: Association d'une activité à une transition

Cette place est en effet une place *substituable* au sens des règles de réduction et les propriétés du réseau ne sont pas modifiées.

Les jetons peuvent être interprétés comme des objets physiques, des informations, des structures de données ou des ressources. Ces entités sont soumises à des événements (franchissements de transitions) qui les font changer d'état (passer d'une place à une autre).

4.2.2 Interaction avec les données et l'environnement.

Lorsque l'on considère un système qui effectue certains traitements, ceux-ci doivent être spécifiés. C'est pourquoi en plus d'un simple nom, il faut associer la spécification des traitements associés soit aux transitions soit aux places. Dans le premier cas on suppose que le traitement sera exécuté de façon ininterrompue lors du franchissement de la transition, dans le second on considère qu'il s'agit d'une activité qui se déroule pendant tout le temps durant lequel la place est marquée.

Le réseau de Petri décrit les enchaînements des traitements à effectuer, il définit la *structure de contrôle* du système modélisé.

Parfois, certains traitements ne sont pas systématiquement exécutés immédiatement après la terminaison des traitements précédents, leurs exécutions peuvent être soumises à des conditions portant sur certaines données (instructions conditionnelles de type "if", par exemple). Un tel phénomène sera décrit par une place avec plusieurs transitions suivantes. C'est, du point de vue de la théorie des réseaux de Petri, une situation de *conflit*. Le franchissement des transitions suivantes est alors soumis à une *condition supplémentaire de franchissement* dont la valeur dépend des données.

Dans le cas de systèmes qui interagissent avec leur environnement, on a également des conditions supplémentaires de franchissement associées aux transitions. Des actions sont associées aux transitions (ou éventuellement aux places). Ces actions (qui sont des traitements particuliers) et ces conditions font intervenir des données ou des événements exté-

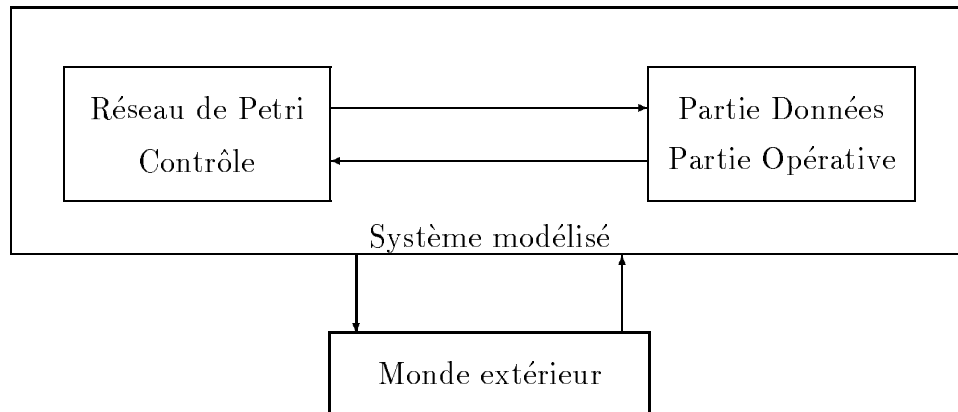


Figure 4.2: Interaction d'un réseau de Petri avec son environnement

rieurs comme des capteurs, des actionneurs, des réceptions ou des émissions de messages. Souvent, l'horloge temps réel joue un grand rôle dans l'environnement et le temps intervient explicitement sous la forme de *temporisation* qui sont des variables booléennes dont la valeur dépend du temps (monostables, chiens de garde etc.).

4.2.3 Contrôle, données et environnement.

Finalement on peut dire que modéliser un système *ouvert* (système communiquant en permanence avec son environnement) à l'aide de réseaux de Petri, revient à le structurer en deux parties.

La première, appelée *partie contrôle*, décrit *tous* les enchaînements potentiels d'événements et d'activités. Cette *partie contrôle* (on dit aussi simplement le *contrôle*) est décrite par le réseau de Petri non interprété.

La seconde, appelée *partie données* (on dit également *partie opérative*) décrit à la fois les structures de données internes au système et les calculs qui sont fait sur ces données, sans spécifier à quels instants ils seront effectués. En plus des données internes, ces calculs peuvent faire intervenir le temps et des informations provenant du monde extérieur. Deux types de calcul peuvent apparaître:

- les conditions,
- les traitements ou actions,

L'interprétation du réseau de Petri spécifie les liens entre la *partie contrôle* d'une part, et la *partie données* et l'environnement du système d'autre part. La figure 4.2 décrit cette structuration. L'interprétation s'exprime sous la forme:

- d'association de conditions aux transitions,
- d'association d'actions aux transitions,
- d'association de traitements aux places,

4.2.4 Prise en compte explicite du temps.

La prise en compte explicite du temps se fait par l'intermédiaire de variables booléennes apparaissant dans les conditions et les actions associées aux transitions et dont la valeur dépend du temps.

Soit $mon(\theta)$ une telle variable. Si on lui affecte la valeur "0", elle reprend automatiquement la valeur "1" au bout d'une durée θ (monostable). Supposons que lors du franchissement d'une transition t_1 , une telle variable soit mise à zéro. Si une transition t_2 , immédiatement sensibilisée après le tir de t_1 possède la condition:

$$(mon(\theta) = 1)$$

alors l'intervalle de temps entre le franchissement de t_1 et celui de t_2 sera exactement de θ .

4.3 Retour sur l'analyse.

4.3.1 Résultats principaux.

Le réseau de Petri ne décrit que la *partie contrôle* du système. Le marquage du réseau de Petri ne fournit donc que l'état du contrôle. L'état du système, c'est-à-dire l'état du réseau de Petri interprété est décrit par le marquage **associé** à l'état des données.

$\begin{aligned} \text{État réseau interprété} &= \text{Marquage} \\ &+ \text{État données} \end{aligned}$
--

$\begin{aligned} \text{État données} &= \text{état variables internes,} \\ &+ \text{temps écoulé} \end{aligned}$
--

D'autre part, il faut remarquer que les évolutions des marquages du réseau de Petri interprété sont des restrictions des évolutions des marquages du réseau avant son interprétation. C'est-à-dire que l'ensemble des marquages accessibles après interprétation est inclus dans l'ensemble des marquages accessibles avant interprétation. En effet, une transition pour pouvoir être franchie au niveau du réseau de Petri interprété doit de toute façon être franchissable au niveau du réseau non interprété. Mais, l'interprétation peut être telle qu'une transition franchissable ne puisse plus être franchie (temps non écoulé, condition portant sur les variables auxiliaires non vérifiée, comportement de l'environnement non compatible). Ceci est illustré par la figure 4.3.

On a donc les résultats suivants:

Réseau non interprété		Réseau interprété
k-borné pour M	\implies	k-borné pour État
f invariant de place	\implies	f invariant de place

Par contre:

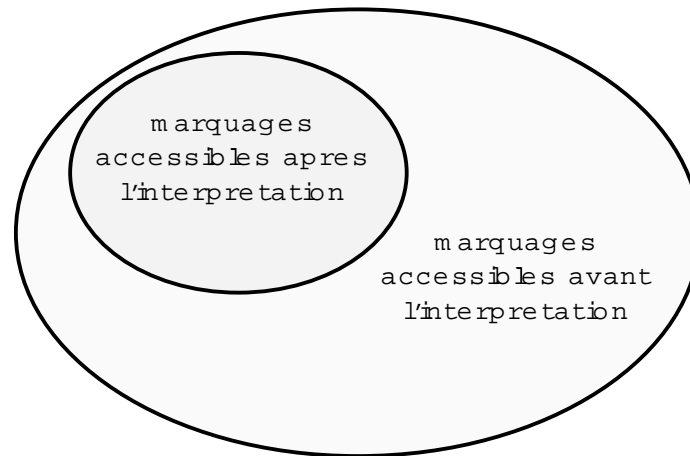


Figure 4.3: Marquages accessibles

Réseau non interprété		Réseau interprété
vivant	\neq	vivant
réinitialisable	\neq	réinitialisable

On peut alors se demander quel est l'intérêt d'analyser un réseau de Petri non interprété puisqu'il sera presque toujours nécessaire de l'interpréter et que ses propriétés peuvent alors être modifiées.

La première raison est que les propriétés des réseaux de Petri interprétés sont *non décidables*. En effet, la complexité devient celle de la preuve formelle de programme. Nous avons vu que malgré tout certaines choses (en particulier les invariants de place) sont conservées, et cela facilitera tout raisonnement ultérieur concernant le réseau interprété.

La seconde raison est que la conception et la validation de la conception d'un système complexe doit se décomposer en étapes. L'étude du réseau de Petri non interprété, plus facile, permettra très tôt de mettre aisément en évidence des incohérences et d'éviter de coûteux retours en arrière. Il ne serait pas *sain* de construire une spécification correcte du point de vue du réseau interprété qui serait fondée sur un réseau non interprété ayant une structure incohérente.

4.3.2 Introduction à divers modèles.

Nous avons vu que les réseaux de Petri interprétés offraient une bonne structuration de la partie contrôle décrite sous la forme d'un réseau de Petri. Par contre, en ce qui concerne la partie données, aucun modèle spécifique n'est offert. Le fait qu'aucune contrainte spécifique n'est imposée peut être un avantage dans certain cas, mais d'un autre côté, aucune méthode de structuration n'est proposée.

Nous allons maintenant voir divers modèles fondés sur les réseaux de Petri qui proposent une structuration de la partie données, en différenciant par exemple les données globales de celles qui sont locales. Par données locales, on entend des données qui n'interviennent

que dans certaines conditions ou actions et qui ne sont accessibles qu'à certains instants. Les réseaux *Colorés*, *Prédicats-Transitions* et *à objets* attachent une partie des données aux jetons. Nous verrons également diverses façons de prendre en compte le temps de manière systématique et rigoureuse en associant des durées aux transitions ou aux places (réseaux de Petri *temporisés*, *temporels* et *stochastiques*).

4.4 Réseau de Petri Colorés.

4.4.1 Repliage d'un réseau de Petri.

La délimitation entre contrôle et données dans un système n'est pas quelque chose d'inhérent au système décrit. Elle résulte d'un choix du concepteur. A la limite, tout le contrôle peut être intégré aux conditions d'exécution des traitements et ainsi disparaître en temps que partie séparée représentable par un réseau de Petri. Inversement, développer le contrôle de façon exagérée peut produire des réseaux de Petri de très grande taille difficilement manipulables.

Une façon de décrire des systèmes relativement complexes avec un réseau de Petri compact est de replier un ensemble de processus ayant la même structure (ou des structures proches les unes des autres) dans une seule composante conservative. L'ennui est que l'on perd l'individualité des processus qui n'est alors connue que dans la partie données. Le marquage des places ne donne que le nombre de processus dans un état donné sans que l'on puisse connaître leurs identités.

L'idée sous-jacente des réseaux de Petri avec individualisation des jetons (c'est-à-dire à la fois les réseaux colorés, les réseaux Prédicats-Transitions et les réseaux à objets) est précisément de pouvoir effectuer ce repliage sans perte d'information et sans perdre la visualisation *graphique* de la structure des processus.

Dans le cas des lecteurs et des écrivains (voir figure 4.4 et figure 4.5), on replie les places $l1, l2, l3$ sur la place $1l, 2l, 2l, 2l$ et $3l, 3l, 3l$ sur $3l$. De même on replie les transitions $d1, d2, d3$ sur $dl, a1, a2, a3$ sur al et $f1, f2, f3$ sur fl . Le réseau devient plus simple puisque les trois processus "*lecteurs*" sont décrits à partir de la même structure (enchaînement de places et de transitions). Malheureusement, lors d'un événement impliquant un lecteur, on ignore s'il s'agit de $l1$, de $l2$ ou de $l3$.

4.4.2 Association de couleurs aux jetons.

Dans le but de différencier les jetons, on leur associe des couleurs (ou des entiers ou des ensembles d'étiquettes). En conséquence, on associe à chaque place l'ensemble des couleurs des jetons qui peuvent y séjourner. A chaque transition on associe un ensemble de couleurs correspondant à des *manières* de franchir cette transition. Dans les cas les plus simples, c'est-à-dire quand tous les processus ont rigoureusement la même structure et sont indépendants les uns des autres, les couleurs des transitions sont directement associées aux processus et l'ensemble des couleurs des places et des transitions sont identiques.

Par exemple, dans le cas des lecteurs et des écrivains, les processus correspondent aux identité des lecteurs. Donc, aux places $1l, 2l$ et $3l$ et aux transitions dl, al et fl on associe

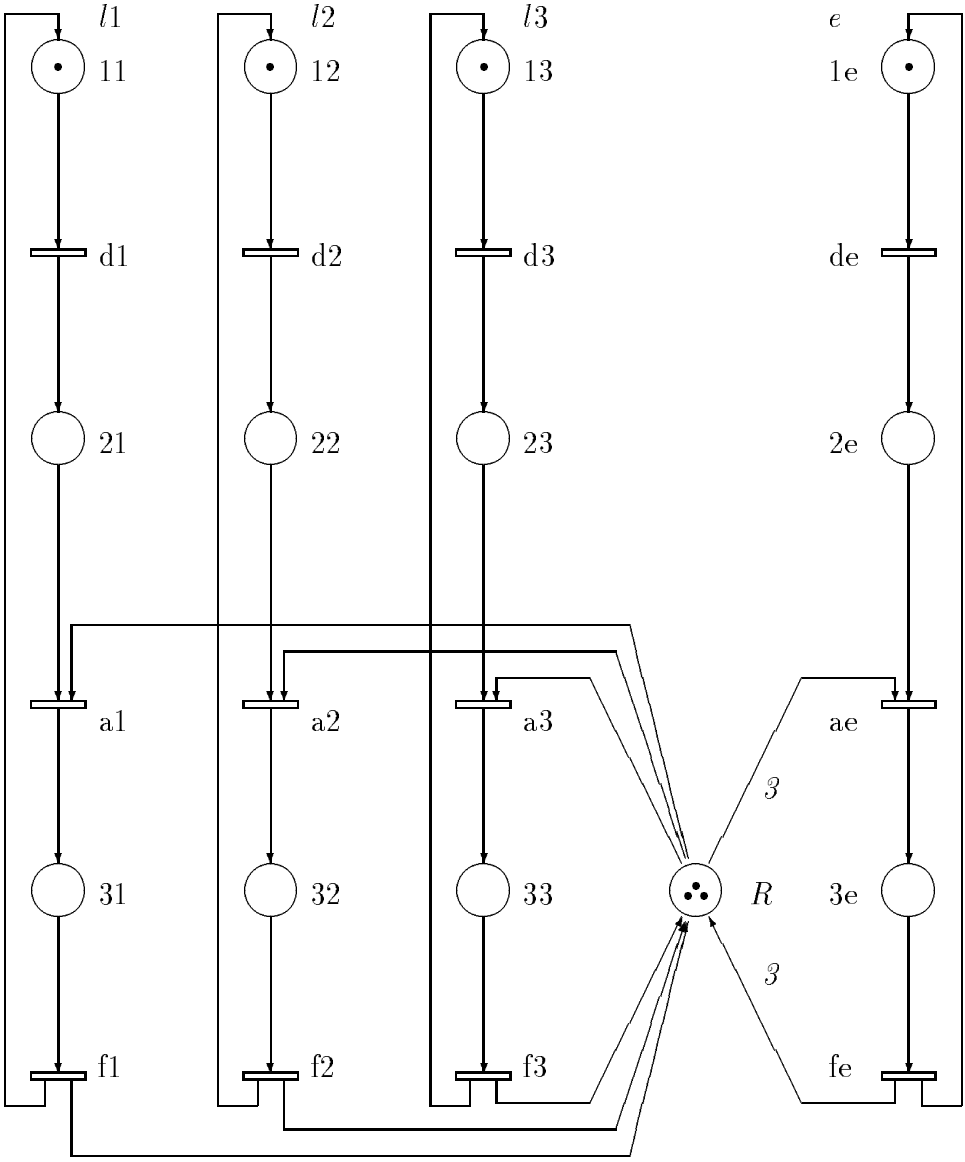


Figure 4.4: Lecteurs et écrivains développés

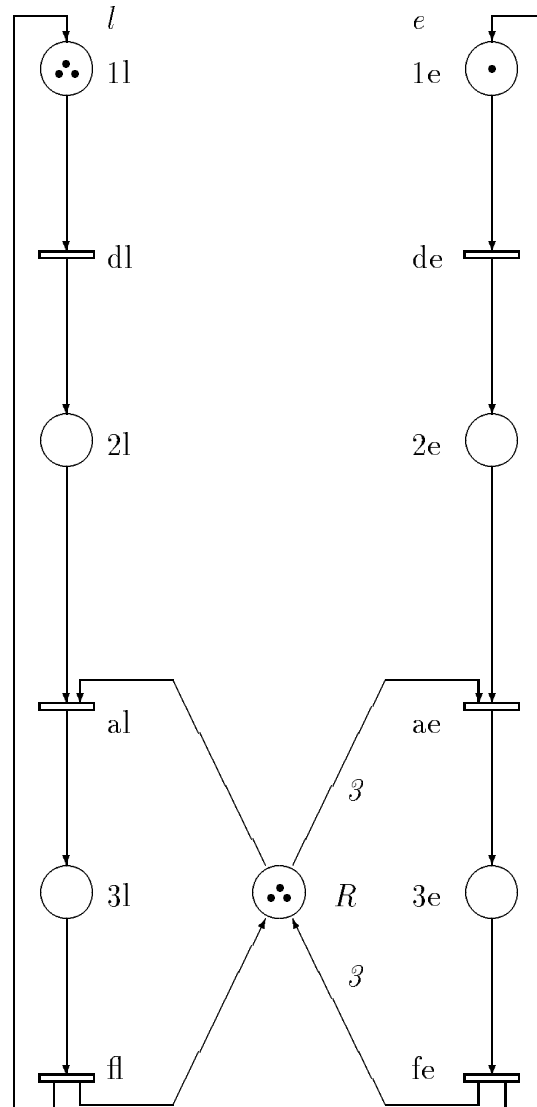


Figure 4.5: Lecteurs et écrivains repliés

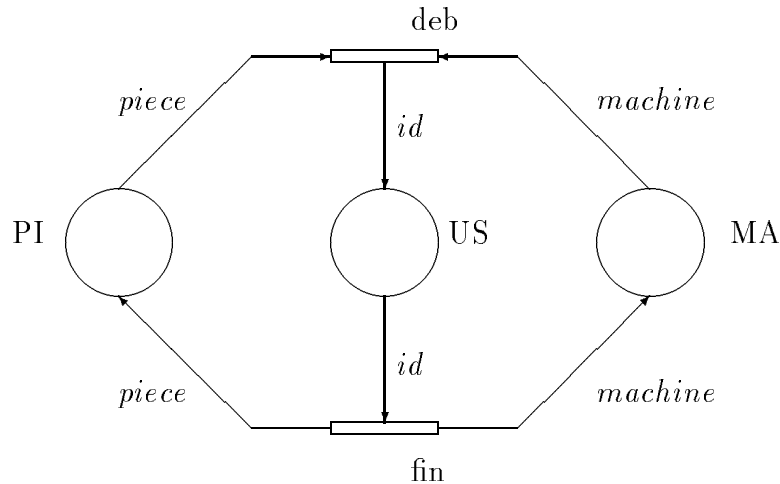


Figure 4.6: Réseau de Petri coloré

l'ensemble:

$$(l1, l2, l3).$$

Ce n'est pas le cas quand des places sont communes à plusieurs processus. Il faut alors introduire des "couleurs composites". Considérons par exemple un atelier de fabrication formé de deux machines et deux pièces à usiner (voir figure 4.6).

La place *PI* possède les couleurs *pi1* et *pi2* qui sont les noms des pièces, la place *MA* les couleurs *ma1* et *ma2* qui sont les noms des machines, par contre, la place *US* contiendra *pi1.ma1*, *pi1.ma2*, *pi2.ma1* et *pi2.ma2* qui correspondent aux diverses opérations d'usinage possibles que nous noterons dorénavant *u11*, *u12*, *u21* et *u22*. Les transitions *deb* et *fin* auront les mêmes couleurs que *US*.

Cela vient du fait que les processus *piece* et *machine* se mélangent dans la place *US*.

4.4.3 Association de fonctions aux arcs.

Dans un réseau coloré, les arcs ne sont plus représentables par de simples entiers. En effet, pour chaque couleur d'une transition (manière de franchir), il faut décrire les actions que l'on va effectuer pour chaque couleur possible dans une place. Chaque arc est donc décrit par une *matrice* dont les lignes sont les façons de franchir les transitions et les colonnes les couleurs possibles dans les places. Ainsi l'arc reliant la place *PI* vers la transition *deb* et celui reliant respectivement *MA* à *deb* sont décrits par:

$$\begin{array}{l}
 u11 \\
 u12 \\
 u21 \\
 u22
 \end{array}
 \begin{array}{cc}
 pi1 & pi2 \\
 \left[\begin{array}{cc}
 1 & 0 \\
 1 & 0 \\
 0 & 1 \\
 0 & 1
 \end{array} \right]
 \end{array}
 \qquad
 \begin{array}{l}
 u11 \\
 u12 \\
 u21 \\
 u22
 \end{array}
 \begin{array}{cc}
 ma1 & ma2 \\
 \left[\begin{array}{cc}
 1 & 0 \\
 0 & 1 \\
 1 & 0 \\
 0 & 1
 \end{array} \right]
 \end{array}$$

En fait, une telle notation serait beaucoup trop lourde et elle est remplacée par une notation fonctionnelle. A un arc est associée une fonction qui à chaque couleur de transition fait correspondre un vecteur ligne d'entiers positifs qui décrit combien de jetons de chaque couleur doivent être retirés à la place quand la transition est franchie d'une certaine manière. La notation la plus simple consiste à écrire ce vecteur sous la forme d'une somme formelle. Par exemple, la fonction *piece* peut s'écrire:

$$\begin{aligned} piece(u_{11}) &= \begin{bmatrix} 1 & 0 \end{bmatrix} = pi_1 \\ piece(u_{12}) &= \begin{bmatrix} 1 & 0 \end{bmatrix} = pi_1 \\ piece(u_{21}) &= \begin{bmatrix} 0 & 1 \end{bmatrix} = pi_2 \\ piece(u_{22}) &= \begin{bmatrix} 0 & 1 \end{bmatrix} = pi_2 \end{aligned}$$

Sur le graphe il suffira de reporter le nom de la fonction. La Figure 4.6 donne un exemple de cette notation qui en fait n'utilise que deux fonctions, la fonction *id* étant la fonction identité (elle ne transforme pas les couleurs). Il faut pourtant bien être conscient qu'en fait on travaille avec des matrices de matrices puisque les éléments des matrices *Pre*, *Post* et *C*, décrits par des fonctions, sont en fait des matrices. Cette façon d'augmenter la compacité des représentations fondées sur les réseaux de Petri exploite donc plutôt la notation matricielle.

Si du point de vue visuel on a effectivement gagné puisque la représentation est très compacte, en fait on continue à avoir la complexité d'un système déplié formé d'un réseau de Petri ordinaire.

4.4.4 Définition formelle (simplifiée).

Un réseau de Petri coloré associé à un marquage initial est un 6-uplet donné par:

$$\mathcal{N}_c = \langle P, T, C_{oul}, C_{sec}, W, M_0 \rangle$$

où:

- P : est un ensemble fini de places,
- T : est un ensemble fini de transitions,
- C_{oul} est un ensemble *fini* de *couleurs*:
- C_{sec} est la fonction *sous-ensemble de couleurs* qui à chaque place et à chaque transition associe un sous-ensemble de C_{oul} (les couleurs possibles pour cette place ou cette transition):

$$C_{sec} : P \cup T \longrightarrow \mathcal{P}(C_{oul})$$

- W est la fonction d'incidence (équivalent de $C = Post - Pre$), chaque élément $W(p, t)$ de W est lui-même une fonction:

$$W(p, t) : C_{sec}(t) \times C_{sec}(p) \longrightarrow N$$

- M_0 est le marquage initial, pour chaque place et pour chaque couleur possible dans cette place, il associe un nombre de jetons:

$$M_0(p) : C_{sec}(p) \longrightarrow N$$

N est l'ensemble des entiers naturels.

4.4.5 Exemple.

Si on revient sur l'exemple de la figure 4.6, nous aurons:

- $C_{out} = \{pi_1, pi_2, ma_1, ma_2, u_{11}, u_{12}, u_{21}, u_{22}\}$
- $C_{sec}(PI) = \{pi_1, pi_2\}$
 $C_{sec}(MA) = \{ma_1, ma_2\}$
 $C_{sec}(US) = C_{sec}(deb) = C_{sec}(fin) = \{u_{11}, u_{12}, u_{21}, u_{22}\}$
- Les fonctions utilisées dans W sont:
 - $id(u_{ij}) = u_{ij}$,
 - $machine(u_{ij}) = ma_j$,
 - $piece(u_{ij}) = pi_i$,

et la matrice W est la suivante:

$$W = \begin{array}{l} PI \\ US \\ MA \end{array} \begin{array}{cc} deb & fin \\ \left[\begin{array}{cc} -piece & piece \\ id & -id \\ -machine & machine \end{array} \right] \end{array}$$

- le marquage initial est:

$$M_0 = \begin{array}{l} PI \\ US \\ MA \end{array} \begin{array}{cccccccc} pi_1 & pi_2 & ma_1 & ma_2 & u_{11} & u_{12} & u_{21} & u_{22} \\ \left[\begin{array}{cccccccc} 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \end{array} \right] \end{array}$$

et il peut s'écrire également sous la forme d'un vecteur de sommes formelles:

$$M_0 = \begin{bmatrix} pi_1 + pi_2 \\ 0 \\ ma_1 + ma_2 \end{bmatrix}$$

4.4.6 Analyse.

Si le nombre de couleurs est fini, il est possible d'énumérer les marquages accessibles. L'analyse par énumération des marquages peut se faire exactement comme pour un réseau de Petri ordinaire. Il faut toutefois remarquer que l'explosion combinatoire est très grande puisqu'il est nécessaire de différencier deux jetons de couleurs différentes. En fait, on travaille sur le réseau de Petri ordinaire obtenu en "dépliant" le réseau coloré.

Lorsque les couleurs et les fonctions présentent une certaine symétrie il est possible de définir des classes d'équivalence entre les marquages accessibles. L'explosion combinatoire est alors réduite et on est plus proche de l'analyse du réseau "sous-jacent" (le réseau ordinaire obtenu en remplaçant les fonctions associées aux arcs par des poids).

En ce qui concerne le calcul des composantes conservatives et répétitives stationnaires, on peut le mener de façon semblable à celle utilisée pour les réseaux de Petri ordinaires. Néanmoins, au lieu de faire des produits d'entiers on doit faire des compositions de fonctions (fonctions de fonctions). Chaque fois que l'on réussit à faire apparaître une ligne (ou une colonne) de zéros dans la matrice W , on obtient une composante. Ce n'est pas toujours possible. D'autre part l'ensemble des composantes ne forment pas nécessairement un espace vectoriel.

Si nous revenons à l'exemple ci-dessus, nous pouvons faire apparaître deux composantes conservatives. En effet, en multipliant la deuxième ligne par *piece* et en remarquant que

$$piece(id) = piece$$

nous obtenons

$$W_1 = \begin{bmatrix} 0 & 0 \\ piece & -piece \\ -machine & machine \end{bmatrix} \begin{bmatrix} PI + piece(US) \\ piece(US) \\ MA \end{bmatrix}$$

et donc nous avons l'invariant de place:

$$M(PI) + piece(M(US)) = M_0(PI) + piece(M_0(US))$$

En multipliant la deuxième ligne par *machine* et en l'ajoutant à la troisième, on obtient l'invariant de place:

$$machine(M(US)) + M(MA) = machine(M_0(US)) + M_0(MA)$$

L'interprétation de ces invariants est simple avec la notation utilisant les sommes formelles. Supposons que l'on ait l'usinage u_{12} en cours, la pièce pi_2 et la machine ma_1 étant en attente. On a $M(PI) = pi_2$, $M(US) = u_{12}$ et $M(MA) = ma_1$. Le premier invariant donne:

$$pi_2 + piece(u_{12}) = pi_1 + pi_2$$

et le second:

$$machine(u_{12}) + ma_1 = ma_1 + ma_2$$

Le premier invariant décrit la conservation des pièces et le second celle des machines.

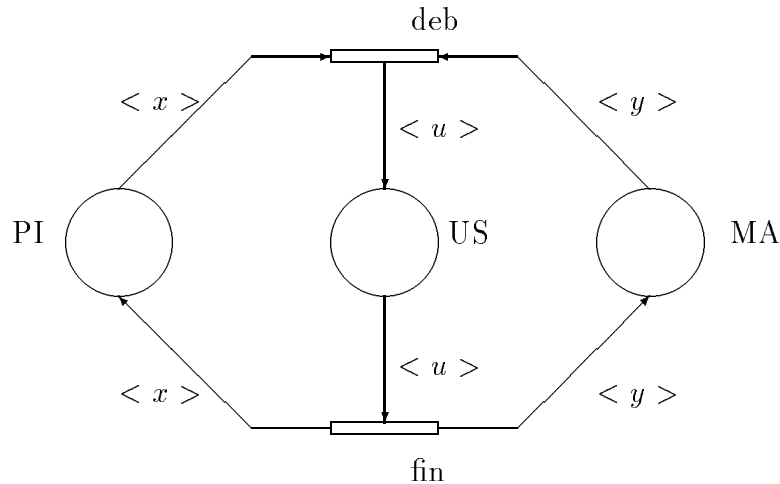


Figure 4.7: Réseau de Petri avec variables formelles.

4.5 Réseaux de Petri Prédicats-Transitions.

4.5.1 Notion de variables, semi-unification.

Nous avons vu que dans les réseaux de Petri Colorés, la puissance de description est augmentée en remplaçant les entiers de la matrice d'incidence d'un réseau de Petri ordinaire par des fonctions. Dans les réseaux Prédicats-Transitions, les transitions d'un réseau de Petri ordinaire sont considérées comme des règles dans un système de logique propositionnelle (sans variables), et la puissance est augmentée en les remplaçant par des règles dans une logique du premier ordre (c'est-à-dire où des variables apparaissent). Une règle (une transition) décrit alors une famille d'événements et non plus seulement un événement. La famille est définie par l'ensemble des substitutions possibles des variables par des valeurs. Au lieu de règles du type:

Si au moins une pièce et au moins une machine faire usinage

On aura des règles du type:

Si une pièce $\langle x \rangle$ et une machine $\langle y \rangle$ faire usinage $\langle u \rangle$.

Ces variables $\langle x \rangle$, $\langle y \rangle$ et $\langle u \rangle$ prendront leur valeurs dans un ensemble de *constants* décrivant les pièces en attente d'usinage et les machines libres. Ces constantes jouent un rôle analogue à celui des couleurs dans les réseaux colorés. Les variables sont associées aux arcs du réseau de Petri à la place des fonctions définies dans les réseaux de Petri colorés. L'exemple vu précédemment sera alors décrit par le réseau de la Figure 4.7.

Il n'est pas nécessaire d'énumérer à priori tous les jetons autorisés dans les places comme nous l'avons fait dans l'exemple pour *PI*, *US* et *MA*. On ignore les différentes façons de

franchir les transitions, car il suffit de savoir que tous les éléments d'une même famille seront traités de la même façon. On applique les règles sur une situation courante et c'est en cherchant toutes les substitutions possibles pour les variables que l'on sait, à un moment donné, de quelles façons les transitions peuvent être franchies. C'est ce mécanisme de substitution des variables par des constantes (c'est-à-dire par des jetons) que l'on appelle l'*unification* (en fait il s'agit de semi-unification car les variables sont toujours directement substituées par des constantes et jamais par d'autres variables).

Quand le mécanisme d'unification produit plus d'une solution on dit (vocabulaire *Intelligence Artificielle*) qu'il y a conflit. L'espace des conflits comprend toutes les transitions franchissables pour un marquage donné, et pour chaque transition l'ensemble des substitutions possibles. Cette notion de conflit n'est pas la même que celle, plus restrictive, concernant des transitions dans un réseau de Petri.

Pour affiner le choix des jetons et des transitions à franchir, on associe des conditions supplémentaires de franchissement aux transitions (par exemple pour spécifier que certaines machines ne peuvent traiter que certaines pièces) comme dans le cas des réseaux de Petri interprétés. Toutefois ces conditions ne portent que sur les constantes associées aux jetons susceptibles d'être déplacés, c'est-à-dire sur les constantes substituées aux variables lors du franchissement des transitions. Ces conditions s'écrivent donc en utilisant les variables formelles des arcs entrant dans la transition. Les actions associées aux transitions s'écrivent de même à l'aide des variables associées aux arcs entrants et sortants.

Remarque:

Dans les réseaux de Petri Colorés le choix spécifique d'un ensemble de jetons est décrit par les fonctions associées aux arcs. Dans les réseaux Prédicats-Transitions, ces choix sont exprimés par des conditions associées aux transitions (sauf le cas simple de l'égalité des constantes qui peut être exprimé en utilisant la même variable associée à plus d'un arc d'entrée d'une transition). Les réseaux de Petri Prédicats-Transitions sont donc plus proches des réseaux de Petri Interprétés de ce point de vue.

4.5.2 Notion de n-uplet de constantes et de variables.

Lors du franchissement de la transition *deb*, on prend une constante représentant une pièce dans la place *PI*, une autre représentant une machine dans la place *MA* et on met dans la place *US* une constante codant le fait que la pièce et la machine sont temporairement en relation. Lors du franchissement de la transition *fin*, signifiant la fin de cette relation, on fait le contraire. Au lieu de coder cette mise en relation temporaire de deux informations, on peut les juxtaposer dans une liste, comme dans un "*record*" en langage Pascal, sauf qu'il s'agit de quelque chose de temporaire, créé et détruit au cours du temps lors de franchissements de transitions. On exprime ainsi une *relation dynamique* entre des constantes.

Ces listes de constantes s'appellent des n-uplets. Ainsi dans la place *US* au lieu de la constante u_{12} on mettra, par exemple, le n-uplet $\langle pi_1, ma_2 \rangle$. Le marquage ne sera plus une distribution de constantes dans les places, ce sera une distribution de n-uplets de constantes. Le n-uplet $\langle pi_1, ma_2 \rangle$ diffère de la somme $\langle pi_1 \rangle + \langle ma_2 \rangle$ puisque dans le premier cas

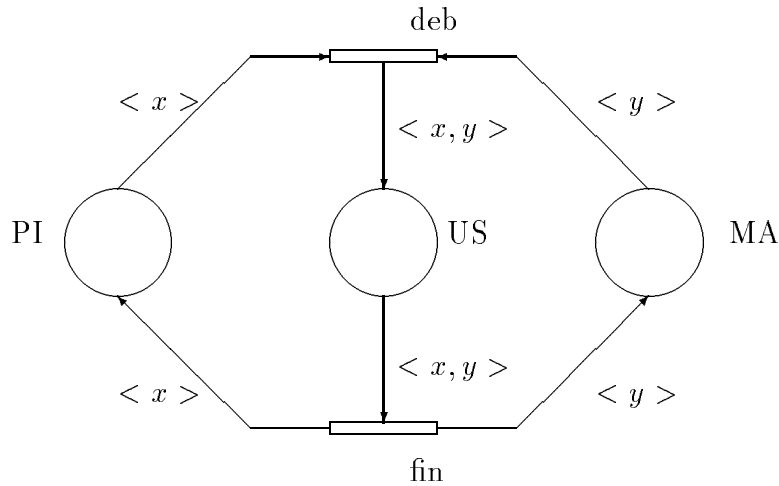


Figure 4.8: Réseau de Petri Prédicats-Transitions.

pi_1 et ma_2 seront utilisés dans des franchissements de transition de façon indépendante alors que le n-uplet devra être substitué à des variables en bloc. C'est pourquoi les expressions associées aux arcs sont également des sommes de n-uplets de variables formelles. On obtient alors la représentation de la figure 4.8.

4.5.3 Définition (simplifiée) des réseaux Prédicats-Transitions.

Un réseau de Petri Prédicats-Transitions initialement marqué est un triplet:

$$\mathcal{N}_{pt} = \langle \mathcal{R}, An, M_0 \rangle$$

- \mathcal{R} est un réseau de Petri ordinaire $\langle P, T, C \rangle$, $C = Post - Pre$,
- An est l'annotation de $calR$, $An = \langle C_{onst}, V, A_{tc}, A_{ta}, A_c \rangle$:
 - C_{onst} est un ensemble de constantes (l'ensemble N par exemple),
 - V est un ensemble de variables formelles qui seront substituées par des constantes de C_{onst} lors des franchissements de transitions,
 - A_{tc} est une application:
$$A_{tc} : T \longrightarrow L_c(C_{onst}, V)$$
associant à chaque transition une condition sous la forme d'un prédicat utilisant à la fois des constantes et des variables formelles,
 - A_{ta} est une application:
$$A_{ta} : T \longrightarrow L_a(C_{onst}, V)$$
associant à chaque transition une action sous la forme d'une suite d'affectation de valeurs à des variables formelles,

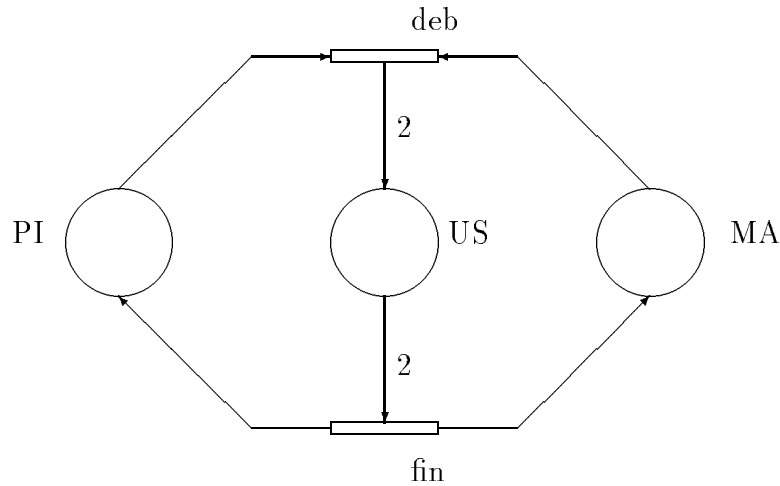


Figure 4.9: Réseau de Petri sous-jacent.

- A_c est une application associant à chaque arc (à chaque élément de C) une somme formelle de n-uplets d'éléments de V telle que son module (la somme de tous les éléments sans les distinguer et en développant les n-uplets) soit égale au poids de l'arc correspondant:

$$\| A_c(p, t) \| = C(p, t)$$

- M_0 est le marquage initial, à chaque place p de P il associe une somme formelle de n-uplets de constantes C_{const} .

Remarques:

La définition des langages L_c et L_a permettant l'expression des conditions et des actions détermine des classes particulières de réseaux de Petri Prédicats-Transitions.

La condition: $\| A_c(p, t) \| = C(p, t)$ exprime le fait qu'il n'y a aucune contradiction entre un réseau de Petri Prédicats-Transitions \mathcal{N}_{pt} et son réseau de Petri *sous-jacent* \mathcal{R} du point de vue du nombre des jetons déplacés. Cela n'est pas le cas pour les réseaux de Petri Colorés où une fonction peut, suivant les cas, "capter" un ou plusieurs jetons sans que cela soit apparent sur le graphe. Dans un tel cas le réseau de Petri sous-jacent obtenu en effaçant toutes les fonctions associées aux arcs peut avoir un comportement différent de celui du réseau Coloré.

4.5.4 Exemple.

Dans la définition des réseaux de Petri Prédicats-Transitions apparaît la notion de réseau sous-jacent qui est obtenu en remplaçant les variables par un poids unitaire et en faisant éclater les n-uplets. La figure 4.9 représente le réseau sous-jacent de la figure 4.8. Ce dernier

représente la structure de contrôle du système alors que le réseau Prédicats-Transitions est une façon structurée pour décrire ensemble contrôle et données.

Avec cet exemple nous avons:

- $C_{const} = \{pi_1, pi_2, ma_1, ma_2\}$

c'est ici un ensemble fini,

- $V = \{x, y\},$

- il n'y a ni condition ni action dans ce cas,

- $A_c = \begin{array}{l} PI \\ US \\ MA \end{array} \begin{array}{cc} deb & fin \\ \left[\begin{array}{cc} - \langle x \rangle & \langle x \rangle \\ \langle x, y \rangle & - \langle x, y \rangle \\ - \langle y \rangle & \langle y \rangle \end{array} \right] \end{array}$

- $M_0 = \begin{array}{l} PI \\ US \\ MA \end{array} \begin{array}{c} \left[\begin{array}{c} \langle pi_1 \rangle + \langle pi_2 \rangle \\ 0 \\ \langle ma_1 \rangle + \langle ma_2 \rangle \end{array} \right] \end{array}$

4.5.5 Représentation par réseau de Petri d'un système de règles.

A partir d'un ensemble de *clauses de Horn* on peut construire un réseau de Petri Prédicats-Transitions. Aux prédicats utilisés dans les clauses on associe des places, et aux clauses on fait correspondre des transitions. Ainsi à la clause

$$A_1 \wedge A_2 \wedge \dots \wedge A_n \longrightarrow B$$

on associe une transition dont les places d'entrée sont A_1, A_2, \dots, A_n et la place de sortie B .

Les constantes pouvant être substituées aux variables de A_i et rendant ce prédicat vrai sont alors décrites par des jetons localisés dans la place correspondante.

Le réseau de Petri obtenu décrit à la fois l'ensemble des clauses et l'ensemble des scénarios d'utilisation de ces clauses (les séquences de franchissement). Les connaissances initiales sont décrites par des transitions "sources" et les conclusions par des transitions "puits". Un scénario efficace est une séquence qui part du marquage vide et se termine par un marquage vide. Chaque conclusion intermédiaire n'est utilisée qu'une fois (les jetons des places d'entrée sont effacés lorsqu'une transition est franchie). Aucune conclusion intermédiaire reste inutilisée (le marquage final est le marquage vide). Il faut remarquer que ce n'est que dans le cas des clauses de Horn qu'on obtient des réseaux de Petri où il est toujours possible de trouver des séquences terminant par le franchissement d'une transition conclusion et produisant des marquages vides.

On peut remarquer qu'un tel scénario correspond à une composante répétitive stationnaire (la séquence est un invariant de transition).

t1	$parent(David, Marie)$	\leftarrow
t2	$parent(Marie, Antoine)$	\leftarrow
t3	$ancetre(x, y)$	$\leftarrow parent(x, y)$
t4	$ancetre(x, z)$	$\leftarrow parent(x, y), ancretre(y, z)$
t5		$\leftarrow ancretre(x, Antoine)$

Figure 4.10: Exemple de système de règles.

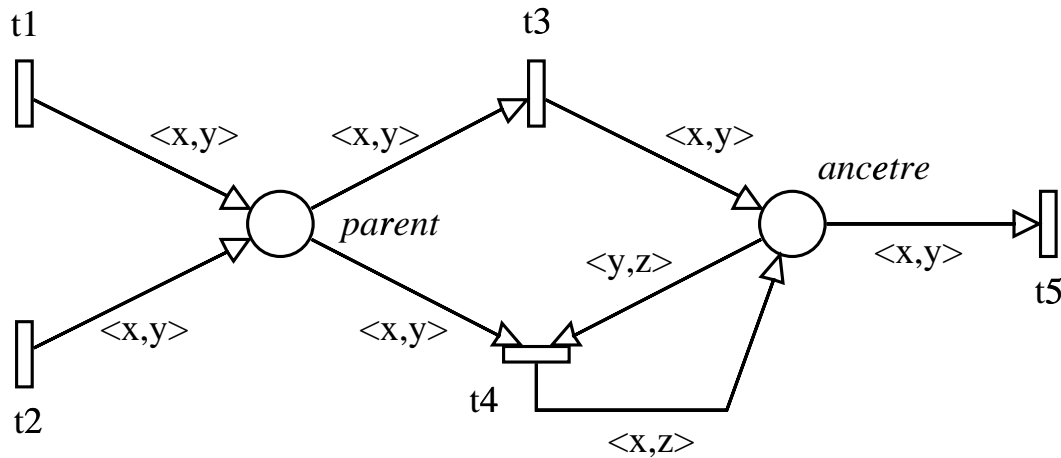


Figure 4.11: Réseau P-T décrivant les règles.

Considérons l'exemple décrit par l'ensemble de règles données par la figure 4.10. La règle **t1** signifie que "David est l'un des parents de Marie"; **t2** que "Marie est l'un des parents d'Antoine"; **t3** que "si x est l'un des parents de y il est également l'un de ses ancêtres" et la règle **t4** que "si y est un ancêtre de z et x l'un des parents de y , alors x est un ancêtre de z ". On cherche à obtenir les ancêtres d'Antoine (règle **t5**).

Cet ensemble de clauses correspond au réseau de Petri de la figure 4.11. Aux transitions **t1**, **t2** et **t5**, il faut ajouter les conditions:

$$\begin{cases} A_{tc}(t1) = (x == \text{"David"}) \wedge (y == \text{"Marie"}) \\ A_{tc}(t2) = (x == \text{"Marie"}) \wedge (y == \text{"Antoine"}) \\ A_{tc}(t5) = (y == \text{"Antoine"}) \end{cases}$$

Aucune action n'est nécessaire.

Une base de composantes répétitives stationnaires du réseau de Petri sous-jacent est:

$$\begin{matrix} t_1 \\ t_2 \\ t_3 \\ t_4 \\ t_5 \end{matrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

Les bons raisonnements correspondront à des combinaisons linéaires des vecteurs de cette base qui passent au moins par l'une des transitions sources **t1** ou **t2** (connaissance initiale) et par la transition conclusion **t5**.

En effet, nous pouvons remarquer que la séquence **t2;t3;t5** produit “Marie est une ancêtre d'Antoine”, et la séquence **t2;t3;t1;t4;t5** produit “David est un ancêtre d'Antoine”.

4.6 Réseaux de Petri à Objets.

4.6.1 Notion d'objet.

Le concept d'objet, de plus en plus utilisé en Génie Logiciel, consiste à structurer une application autour d'entités encapsulant à la fois des structures de données, sous la forme d'une liste d'attributs, et des méthodes de transformation de ces données. Cette approche s'oppose à la démarche fonctionnelle suivant laquelle on définit de façon séparée des structures de données et des fonctions que le système doit remplir.

Une *classe d'objets* est définie par un ensemble d'*attributs* (également appelés *propriétés*) et un ensemble d'*opérations* (on dit également *méthodes*) permettant de manipuler les valeurs des attributs. On peut définir une classe à partir d'une autre classe par *héritage*. La nouvelle classe “hérite” des définitions d'attributs et d'opérations de la classe définie au préalable et on peut compléter ces définitions par des attributs et des opérations spécifiques. On peut parler de *sous-classe*.

Les classes ne sont que des définitions. Un objet particulier est une *instance de classe d'objets*. On peut affecter des valeurs aux attributs de tels objets et exécuter leurs opérations. L'encapsulation confère aux objets une certaine autonomie et une certaine rémanence. Un objet *naît* (instanciation) et *disparaît* (destruction) dynamiquement pendant l'exécution du logiciel, mais ces événements doivent être rares vis-à-vis des événements correspondant à l'exécution des opérations sur les attributs.

Les réseaux à Objets peuvent être considérés comme une utilisation des réseaux de Petri Prédicats-Transitions dans le contexte d'une approche à objets. Les jetons ne sont plus vus comme des constantes, mais comme des n-uplets d'instances de classes d'objets. En plus des attributs définis par la classe, un attribut *implicite* contient le nom de la place où l'objet est localisé. Les opérations sont associées aux transitions; elles portent sur les attributs des objets situés dans les places d'entrée. Une opération associée à une transition *t* ne pourra être exécutée pour un objet que si celui-ci est localisé dans une place d'entrée de *t*.

D'un certain point de vue cette approche est moins structurante que l'approche à objet classique puisque une opération est définie dans le cadre d'un ensemble de classes d'objets (celles des objets pouvant être dans les places d'entrées) et non d'une seule classe. D'un autre point de vue elle est plus structurante car elle introduit une notion de *contrôle*. Pour qu'une opération soit applicable sur les attributs d'une instance d'objet, il est nécessaire que celle-ci soit dans un certain *état* c'est-à-dire qu'il soit dans une certaine *place*.

Dans l'exemple que nous considérons, les objets sont associés directement aux objets physiques. Nous avons les pièces et les machines. Chacun de ces objets possède un certain nombre d'attributs comme:

- le nom de l'instance,
- la date de livraison pour une pièce,
- l'opération à faire pour une pièce,
- la liste des opérations pouvant être faites pour une machine.

Les conditions et les actions associées aux transitions correspondent aux méthodes et travaillent sur les attributs des objets substitués aux variables formelles lors du franchissement des transitions.

4.6.2 Définition (simplifiée) des réseaux de Petri à Objets.

Un réseau de Petri à Objets est défini par le 9-uplet:

$$\mathcal{N}_o = \langle P, T, C_{lass}, V, Pre, Post, A_{tc}, A_{ta}, M_0 \rangle$$

- P est un ensemble fini de places,
- T est un ensemble fini de transitions,
- C_{lass} est un ensemble fini de classes d'objets, éventuellement organisé en une hiérarchie et définissant pour chaque classe un ensemble d'attributs,
- V est un ensemble de variables typées par C_{lass} ,
- Pre est la fonction *place précédente* qui à chaque arc d'entrée d'une transition fait correspondre une somme formelle de n-uplets d'éléments de V ,
- $Post$ est la fonction *place suivante* qui à chaque arc de sortie d'une transition fait correspondre une somme formelle de n-uplets d'éléments de V ,
- A_{tc} est une application qui à chaque transition associe une condition faisant intervenir les variables formelles associées aux arcs d'entrée et les attributs des classes correspondantes,
- A_{ta} est une application qui à chaque transition associe une action faisant intervenir les variables formelles associées aux arcs d'entrée et les attributs des classes correspondantes,
- M_0 est le marquage initial qui associe à chaque place une somme formelle de n-uplets d'instances d'objets (les objets doivent être représentés par des identificateurs, leur nom par exemple).

4.6.3 Exemple.

En considérant toujours le même exemple, le réseau n'est pas modifié du point de vue de sa représentation graphique et est donc représenté Figure 4.8.

Définissons les divers éléments:

- l'ensemble des classes est: $C_{lass} = \{piece, machine\}$ avec:

$$- machine = \begin{cases} nom : \text{identificateur} \\ operations : \text{liste des opérations possibles} \\ maintenance : \text{date d'arrêt pour maintenance} \end{cases}$$

$$- piece = \begin{cases} nom : \text{identificateur} \\ operation : \text{opération à faire} \\ date : \text{date de livraison} \end{cases}$$

- l'ensemble des variables est:

- x de type *piece*,
- y de type *machine*,

- les matrices *Pre* et *Post* sont:

$$Pre = \begin{bmatrix} \langle x \rangle & 0 \\ 0 & \langle x, y \rangle \\ \langle y \rangle & 0 \end{bmatrix} \quad Post = \begin{bmatrix} 0 & \langle x \rangle \\ \langle x, y \rangle & 0 \\ 0 & \langle y \rangle \end{bmatrix}$$

- un exemple de condition peut être:

$$A_{tc}(deb) = x.operation \in y.operations$$

- le marquage initial est défini par:

- pi_1, pi_2 sont de classe *piece*,
- ma_1, ma_2 sont de classe *machine*,

$$- M_0 = \begin{bmatrix} \langle pi_1 \rangle + \langle pi_2 \rangle \\ 0 \\ \langle ma_1 \rangle + \langle ma_2 \rangle \end{bmatrix}$$

4.6.4 Analyse.

Les réseaux de Petri Prédicats-Transitions et à Objets sont définis sous la forme de réseaux de Petri ordinaires (les réseaux sous-jacents) munis d'inscriptions. L'analyse portera d'abord sur ces réseaux. On procède donc comme pour des réseaux de Petri interprétés.

Les invariants du réseaux de Petri sous-jacents donnent des résultats sur la conservation des *nombres* d'objets indépendamment de leurs classes. Pour obtenir une information plus précise, se rapprochant de celle que l'on obtient avec les réseaux de Petri Colorés, on procède

par *projections*. Faire une projection c'est, en quelque sorte, construire un réseau de Petri sous-jacent ne concernant qu'une classe d'objet (ou un ensemble de classes).

Si on choisit de projeter par rapport à une classe $class_i$ de C_{class} , on remplace dans Pre et $Post$ toutes les variables de classe $class_i$ (ou d'une sous-classe de cette classe) par "1" et toutes les autres par "0". On obtient alors une matrice d'incidence $C = Post - Pre$ d'entiers naturels et on peut calculer des composantes comme dans un réseau de Petri ordinaire. Quand on écrit un invariant de place, ce n'est pas le marquage des places qu'il faut considérer mais leur projection c'est-à-dire le *nombre* d'objets de la classe $class_i$ contenus dans la place.

Revenons à notre exemple, et supposons que l'on effectue une projection par rapport à la classe *piece*, on remplacera x par "1" et y par "0". On obtient la matrice:

$$\begin{array}{cc} & \begin{array}{cc} deb & fin \end{array} \\ \begin{array}{c} PI \\ US \\ MA \end{array} & \left[\begin{array}{cc} -1 & 1 \\ 1 & -1 \\ 0 & 0 \end{array} \right] \end{array}$$

La projection par rapport à la classe *machine* donne:

$$\begin{array}{cc} & \begin{array}{cc} deb & fin \end{array} \\ \begin{array}{c} PI \\ US \\ MA \end{array} & \left[\begin{array}{cc} 0 & 0 \\ 1 & -1 \\ -1 & 1 \end{array} \right] \end{array}$$

Les invariants de place sont alors:

$$\| M(PI) \|_{piece} + \| M(US) \|_{piece} = Constante$$

qui correspond au vecteur

$$f_{piece} = \left[\begin{array}{ccc} 1 & 1 & 0 \end{array} \right]$$

et:

$$\| M(US) \|_{machine} + \| M(MA) \|_{machine} = Constante$$

qui correspond au vecteur

$$f_{machine} = \left[\begin{array}{ccc} 0 & 1 & 1 \end{array} \right]$$

On retrouve de cette façon des résultats analogues à ceux que l'on avait obtenus avec les réseaux de Petri Colorés.

4.7 Réseaux de Petri temporisés.

Un réseau de Petri ordinaire décrit une relation de *causalité* entre des événements. Un événement a est la cause de b , a précède toujours b , a et b sont ordonnés dans le temps. Le temps est pris en compte de manière qualitative. Nous allons voir maintenant des approches permettant de prendre en compte le temps de façon quantitative. Dans les modèles qui vont être présentés, le temps est directement associé au réseau de Petri. D'un certain point de vue il fait partie du *contrôle* au lieu d'être rejeté dans la *partie données* de façon non structurée.

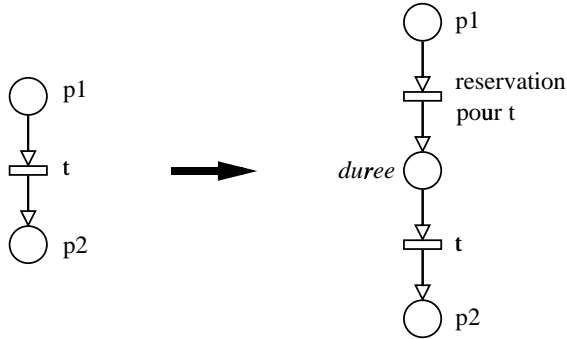
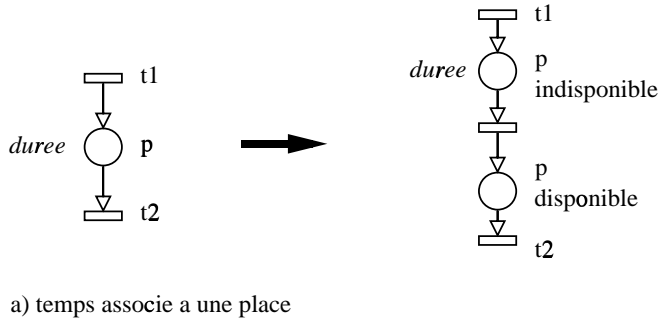


Figure 4.12: Temporisation d'un réseau de Petri

4.7.1 Temps associé à une place:

Pour une place représentant une activité, il s'agit simplement de noter la durée θ de cette activité. On peut considérer qu'en fait la place concernée éclate en une séquence "place-transition-place". La première place correspond à l'activité *en cours*, la transition correspond à l'événement *temps écoulé* et la dernière place correspond à une attente éventuelle (synchronisation avec d'autres activité) après la fin de l'activité. Ce schéma est représenté dans la figure 4.12. Pendant que l'activité est en cours le jeton ne peut être utilisé pour franchir une transition. On dit qu'il est *non disponible*. Une fois l'activité terminée, le jeton devient disponible et la transition $t2$ peut éventuellement être franchie.

4.7.2 Temps associé à une transition:

Cette association n'a de sens que si la transition est interprétée comme une activité et non comme un événement instantané. Comme nous l'avons mentionné plus haut, on suppose implicitement que cette activité est ininterrompible vis-à-vis du comportement global du système modélisé ou bien qu'il s'agit d'une simple abréviation.

On peut considérer qu'en fait cette transition est éclatée en une séquence "transition place transition". La première transition correspond à l'événement *instantané* de début d'activité (on enlève tous les jetons), la place sert à mémoriser l'activité en cours et la dernière transition correspond à l'événement *instantané* de fin d'activité (on met les jetons dans les places de sortie).

Cette équivalence est décrite par la figure 4.12. Le franchissement de la première transition suivant la place $p1$ de la forme éclatée correspond à la réservation des jetons (les jetons *réservés* ne peuvent plus être utilisés pour franchir une autre transition que t . Après le franchissement de la deuxième transition les jetons sont libérés.

Que ce soit dans ce cas, comme dans le cas précédent, on a en fait des jetons qui *disparaissent* ou qui sont *indisponibles* ou *réservés* pendant un certain temps, puis qui réapparaissent dans le marquage. La figure 4.12 montre que les deux approches sont en fait équivalentes.

4.7.3 Définition.

Un réseau de Petri temporisé est une paire $\mathcal{N}_t = \langle \mathcal{N}, \Theta_f \rangle$ avec:

- \mathcal{N} est un réseau de Petri $\langle P, T, Pre, Post \rangle$ avec un marquage initial M_0 ,
- Θ_f est la fonction *durée de franchissement*:

$$\Theta_f : T \longrightarrow \mathcal{Q}^+$$

qui à chaque transition fait correspondre un nombre rationnel positif décrivant la durée du franchissement.

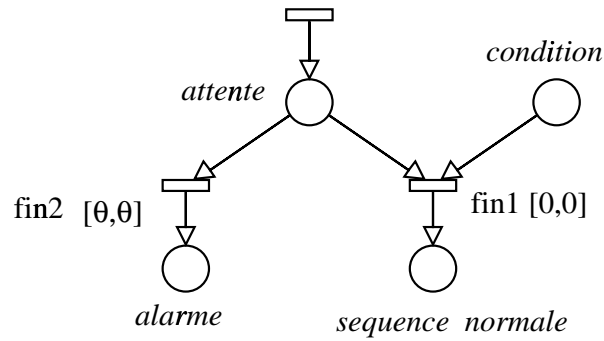


Figure 4.13: Chien de garde

4.8 Réseaux de Petri temporels.

4.8.1 Représentation d'un chien de garde.

Certains mécanismes impliquent en fait qu'un jeton donné ne soit visible que de certaines transitions à un instant donné. Un cas typique est le *chien de garde*. Considérons la Figure 4.13. La place *attente* permet d'être réceptif (franchissement de la transition *fin1*) à l'arrivée d'un événement (condition vraie) sous la forme d'un jeton dans la place *condition*. Mais si le jeton n'est toujours pas présent au bout d'un temps θ , alors on décide de lancer une alarme (franchissement de la transition *fin2*). Un temps d'activité associé à la place *attente* retarderait le franchissement de *fin1* même si la place *condition* contient un jeton plus tôt. Un temps d'activité associé à la transition *fin2* ferait que le jeton contenu dans la place *attente* serait immédiatement pris de façon irréversible et que l'alarme serait alors déclenchée au bout du temps θ même si un jeton était arrivé entre temps dans la place *condition*.

La solution est d'introduire une durée de sensibilisation $\theta_s(t)$ à chaque transition. Une transition t n'est franchissable que si elle est restée sensibilisée durant $\theta_s(t)$. La différence vis-à-vis de la durée de franchissement est que pendant toute cette durée les jetons sont disponibles dans les places d'entrée de t et peuvent éventuellement être utilisés par une transition en conflit avec t .

Si l'on revient à l'exemple ci-dessus, il suffit d'associer à la transition *fin2* une durée θ_s égale à la valeur du chien de garde et à *fin1* une durée de sensibilisation immédiate.

Le domaine principal d'utilisation de ce modèle (validation des protocoles de communication) a fait que, dès le départ, il a été défini une durée *imprécise* de sensibilisation de t sous la forme d'un intervalle $i(t) = [\theta_{s,min}(t), \theta_{s,max}(t)]$. La transition t ne peut être franchie avant que $\theta_{s,min}(t)$ ne soit écoulé. Elle doit impérativement être franchie avant que la durée de sensibilisation ne dépasse $\theta_{s,max}(t)$. Toute les valeurs de $\theta_s(t)$ telles que:

$$\theta_{s,min}(t) \leq \theta_s(t) \leq \theta_{s,max}(t)$$

sont des durées de sensibilisation possibles pour t .

4.8.2 Définition.

Un réseau de Petri *temporel* est une paire $\mathcal{N}_{tl} = \langle \mathcal{N}, I \rangle$ où:

- \mathcal{N} est un réseau de Petri $\langle P, T, Pre, Post \rangle$ muni d'un marquage initial M_0 ,
- I est une fonction qui à chaque transition t fait correspondre un intervalle fermé rationnel

$$i(t) = [\theta_{s,min}(t), \theta_{s,max}(t)],$$
qui décrit une *durée de sensibilisation*.

4.8.3 Comparaison avec les autres modèles.

Les réseaux de Petri temporels sont plus généraux que les réseaux temporisés. Ils permettent de décrire les chiens de garde ce que les réseaux temporisés ne permettent pas. Par contre, pour passer d'un réseau temporisé à un réseau temporel, il suffit de remplacer chaque transition t du premier par une séquence $[t_1; t_2]$ avec :

- $\theta_{s,min}(t_1) = \theta_{s,max}(t_1) = 0$
- et $\theta_{s,min}(t_2) = \theta_{s,max}(t_2) = \theta(t)$

D'autre part les réseaux de Petri ordinaires (sans explicitation du temps) correspondent simplement au cas où tous les intervalles de sensibilisation sont égaux à $[0, \infty]$.

4.8.4 Analyse

Pour savoir si à un instant donné une transition sensibilisée peut être franchie ou non, il faut savoir si sa durée de sensibilisation est supérieure à $\theta_{s,min}$ et inférieure à $\theta_{s,max}$. Pour cela il est nécessaire d'associer à chaque transition sensibilisée un horloge locale (un chronomètre) qui va compter le temps à partir de l'instant de sa sensibilisation.

L'état d'un réseau de Petri temporel est donc défini par le marquage courant et par la valeur de des horloges associées aux transitions sensibilisées. Il y a deux types de changements d'états. Le premier est lié à l'écoulement du temps lorsque aucune transition n'est franchie. Si $\delta\theta$ s'est écoulé, alors, sans changer le marquage courant il faut ajouter $\delta\theta$ à chacune des horloges associées aux transitions sensibilisées. Le deuxième type de changement d'état correspond au franchissement d'une transition. Comme le franchissement est instantané, le temps (en temps que variable continue) n'évolue pas. A temps constant, on modifie le marquage, on construit la nouvelle liste des transitions sensibilisées. Pour les transitions qui était sensibilisées avant le franchissement on laisse la valeur de leur horloge inchangée. Pour chaque transition t nouvellement sensibilisées on initialise son horloge à la valeur du temps courant Θ . Les dates au plus tôt et au plus tard de franchissement pour t seront alors $\Theta + \theta_{s,min}(t)$ et $\Theta + \theta_{s,max}(t)$.

Il est clair que l'ensemble des états est infini, même si le réseau de Petri sous-jacent (le réseau de Petri ordinaire à partir duquel le réseau de Petri temporel est construit) est borné pour le marquage initial choisi. En effet, pour un marquage donné il y a une infinité d'états à cause de l'évolution continue du temps.

Toutefois, l'évolution du temps sans qu'une transition ne soit franchie n'a pas d'influence sur les évolutions futures (du moins lorsque l'on ne sort pas des intervalles de franchissement

pour les transitions franchissables). L'analyse des réseaux de Petri temporels est donc fondée sur une notion de classe d'états qui regroupe tous les états ayant même marquage et n'étant différents que par la valeur des horloges associées aux transitions sensibilisées. Tous les états regroupés dans une classe doivent être tels que l'évolution future du réseau de Petri temporel est la même (l'ensemble des états accessibles doit être le même).

Cette notion de classe est encore insuffisante pour obtenir un ensemble fini caractérisant tous les états possibles. En effet le temps évoluant sans cesse, même si le réseau de Petri retrouve une configuration similaire du point de vue du marquage et des intervalles pendant lesquels les transitions pourront être franchies, la valeur de Θ sera différente. Pour résoudre ce problème il faut passer en temps relatif et donc considérer que l'origine des temps est toujours la date du dernier franchissement effectué. On doit donc faire un changement de l'origine des temps lors de chaque franchissement. Pour les transitions nouvellement sensibilisées, l'intervalle est alors simplement :

$$[\theta_{s,min}(t), \theta_{s,max}(t)].$$

Pour une transition t_i qui était auparavant sensibilisée, si son intervalle était :

$$[\theta_{min}(t_i), \theta_{max}(t_i)]$$

et si la date du franchissement de la transition t_j est $\theta_f(t_j)$ alors le nouvel intervalle de t_i devient :

$$[\theta_{min}(t_i) - \theta_f(t_j), \theta_{max}(t_i) - \theta_f(t_j)].$$

Le problème vient du fait que l'on cherche à caractériser tous les comportements quelle que soit la date de franchissement de t_j pendant tout son intervalle de franchissement autorisé. Il faut donc considérer toutes les valeurs de $\theta_f(t_j)$ comprises entre $\theta_{f,min}(t_j)$ et $\theta_{f,max}(t_j)$. L'intervalle à prendre en compte est donc :

$$[\min(0, \theta_{min}(t_i) - \theta_{f,max}(t_j)), \theta_{max}(t_i) - \theta_{f,min}(t_j)].$$

Remarque Cette approche permet de recouvrir tous les cas possible, mais elle peut ajouter des comportements supplémentaires car les intervalles de franchissements de toutes les transitions sensibilisées avant et après le franchissement de t_j sont tous agrandis. Il faut donc ajouter des contraintes temporelles sur les écarts entre les dates de franchissement de ces transitions pour caractériser exactement l'ensemble des états accessibles.

4.8.5 Exemple

Pour simplifier, nous ne donnons ici que les contraintes sur le franchissement des transitions (et non celle sur les écarts entre les franchissements). Le réseau de Petri temporel est celui de la figure 4.14.

Le graphe des classes est celui de la figure 4.15. Nous voyons que le franchissement de la transition d dans l'intervalle $[1, 2]$ introduit de l'imprécision sur la date de franchissement de b à cause du changement de l'origine du temps.

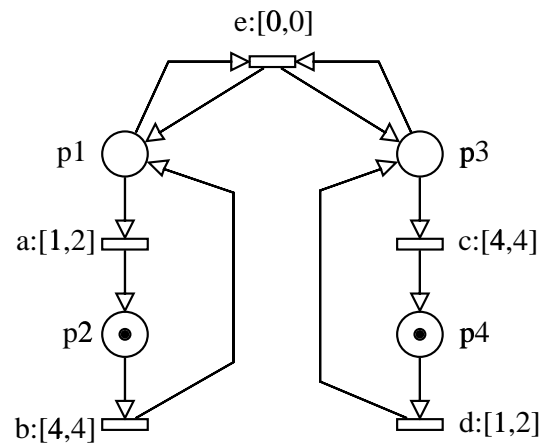


Figure 4.14: Réseau de Petri t-temporel

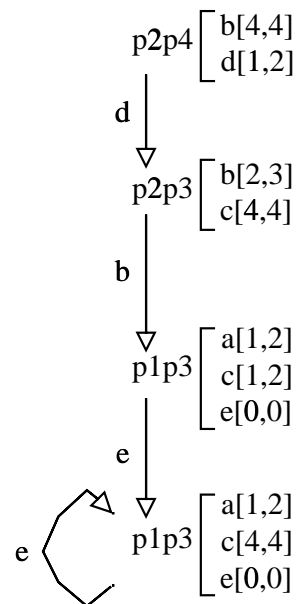


Figure 4.15: Graphe de classes associé

4.9 Réseaux de Petri stochastiques.

4.9.1 Limite des réseaux temporisés et temporels pour l'analyse.

L'utilisation des réseaux de Petri temporisés permet de construire des modèles réalistes pour évaluer des systèmes à événements discrets par *simulation*. En effet, rien n'empêche de modifier la fonction Θ_f pour associer aux transitions non pas une valeur constante mais le résultat d'un tirage aléatoire. Néanmoins, nous avons vu sur un exemple que les propriétés d'un réseau temporisé pouvaient être différentes de celles du réseau de Petri sous-jacent.

L'utilisation des réseaux de Petri temporels permet la description aisée des chiens de garde et la construction d'un graphe de couverture des états accessibles dans les cas simples où le réseau de Petri sous-jacent est borné et où les intervalles de temps associés aux transitions sont donnés sous la formes d'entiers. Ce modèle a servi à vérifier formellement un certain nombre de protocoles, néanmoins la taille du graphe de couverture explose très vite lorsque les intervalle de temps sont trop divers.

Le problème principal vient du fait que la description de l'état de ces réseaux doit comprendre non seulement le marquage mais aussi des informations temporelles. Pour pouvoir utiliser la puissance de l'analyse *Markovienne*, il faut que les systèmes soient sans mémoire du passé, c'est-à-dire que si un événement produit un franchissement de transition t et transforme le marquage M_1 en M_2 , l'évolution future des transitions qui étaient sensibilisées par M_1 avant le franchissement de t doit être identique à celle qu'elles subiraient si elles venaient juste d'être sensibilisées par M_2 .

Seules les distributions *geometriques* et *exponentielles* vérifient ce fait. Les réseaux de Petri *stochastiques* sont donc des réseaux de Petri où les *durées de sensibilisation* associées aux transitions sont définies par de telles distributions afin de pouvoir contruire un processus Markovien équivalent et ainsi analyser les comportements du réseau.

4.9.2 Durée de sensibilisation stochastique.

Au lieu de prendre, de façon égale, toutes les valeurs d'un intervalle ($[\theta_{min}, \theta_{max}]$) comme dans les réseaux de Petri temporels, on peut supposer que la durée de sensibilisation est une variable stochastique θ avec une distribution de probabilité exponentielle:

$$Pr[\theta \leq t] = 1 - e^{-\lambda t}$$

La fonction $Pr_\theta(t)$ décrit la probabilité pour que le franchissement ait lieu avant t c'est-à-dire pour que la durée de sensibilisation soit inférieure à t .

On en déduit que la valeur moyenne de la durée de sensibilisation est

$$\bar{\theta} = \int_0^\infty (1 - Pr_\theta(t)).dt = \int_0^\infty e^{-\lambda \cdot t}.dt = \frac{1}{\lambda}$$

λ est appelé *taux de transition*.

4.9.3 Définition des réseaux de Petri stochastiques.

Un réseau de Petri *stochastique* est une paire $\mathcal{N}_{ts} = \langle \mathcal{N}, \Lambda \rangle$

- \mathcal{N} est un réseau de Petri avec un marquage initial,
- Λ est une fonction qui à chaque transition t associe un taux de transition $\lambda_t = \Lambda(t)$

Remarque.

Cela revient à associer à chaque transition un intervalle de sensibilisation continu $[0, \infty[$ avec une distribution exponentielle. Ceci explique que l'ensemble des marquages accessibles est le même que celui du réseau sous-jacent.

4.9.4 Obtention d'un processus de Markov.

Soit deux marquages accessibles M_i et M_j tels qu'une seule transition t_k soit telle que:

$$M_i \xrightarrow{t_k} M_j$$

le taux de transition de l'état M_i vers l'état M_j est bien évidemment $\Lambda(t_k)$ (la loi de probabilité étant exponentielle).

Si deux transitions t_k et t_m permettent de passer de M_i à M_j , alors le taux de transition de M_i vers M_j sera " $\Lambda(t_k) + \Lambda(t_m)$ ".

A partir d'un réseau de Petri stochastique $\mathcal{N}_{ts} = \langle \mathcal{N}, \Lambda \rangle$, on peut construire un processus de Markov. Les états (espace d'état discret et temps continu) seront l'ensemble des marquages accessibles de \mathcal{N} : $A(\mathcal{R}, M_0)$. A partir de la fonction Λ on écrit directement la matrice Q des taux de transition. La colonne j de Q décrit l'évolution de la probabilité π_j du marquage M_j au cours du temps. Le terme q_{ij} est le taux de transition pour arriver dans l'état M_j à partir de l'état M_i . Le terme q_{jj} est négatif car il décrit le taux de transition permettant de quitter l'état M_j .

Le régime stationnaire du processus Markovien sera un vecteur Π^* tel que:

$$\Pi^{*T} Q = 0 \quad \text{avec} \quad \sum_i \pi_i^* = 1 \quad (4.1)$$

Si B est une propriété qui est vérifiée par un ensemble de marquages EM , on a:

$$Pr_B = \sum_{M_i \in EM} \pi_i \quad (4.2)$$

L'équation 4.2 permet, par exemple, le calcul de la valeur moyenne du nombre de jetons contenus dans une place donnée.

On peut faire les remarques suivantes:

- On peut considérer des taux de transition dépendant du marquage des places d'entrée, le processus stochastique restera Markovien.

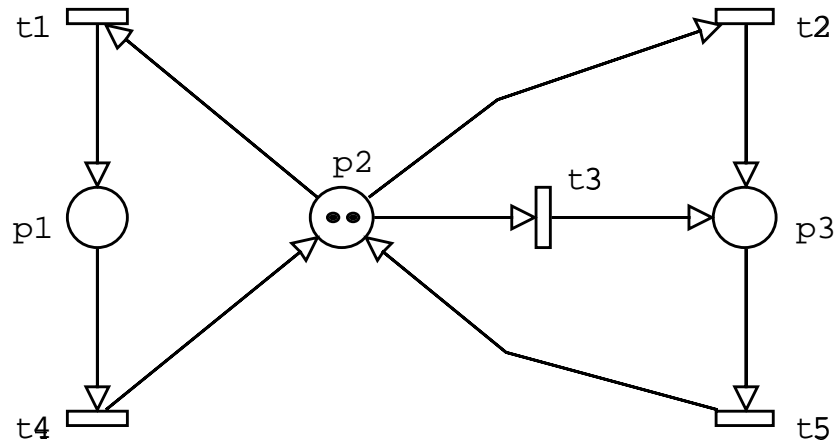


Figure 4.16: Réseau de Petri stochastique.

- Lorsque le réseau de Petri est réinitialisable (graphe des marquages accessible fortement connexe), l'équation 4.1 possède une solution unique.
- L'ensemble des places appartenant à une composante conservative positive est analogue à une sous-chaîne de Markov fermée.
- Une place non bornée correspond à une chaîne ouverte. Pour des problèmes de complexité, on se restreint généralement à des réseaux de Petri bornés ou ne possédant qu'une ou deux places non bornées pour leur marquage initial.
- Une équation particulière j du système 4.1 est décrite par la colonne j de la matrice Q . Elle correspond à l'application du théorème des coupes autour du nœud correspondant au marquage M_j . En régime stationnaire, la dérivée de la probabilité de ce marquage est nulle.

4.9.5 Exemple.

Soit le réseau de Petri stochastique de la figure 4.16. Supposons que les taux de transition sont les suivants:

$$\begin{aligned}\Lambda(t_i) &= \lambda_i \text{ pour } i = 1 \text{ à } 4 \\ \Lambda(t_5) &= M(p_3) \cdot \lambda_5\end{aligned}$$

Le graphe des marquages accessibles est donné par la figure 4.17 et la matrice des taux de transition est donné par la figure 4.18 avec les six marquages suivants:

$$\begin{aligned}M_0 &= p_2^2 & M_1 &= p_2 \cdot p_3 \\ M_2 &= p_3^2 & M_3 &= p_1 \cdot p_2 \\ M_4 &= p_1^2 & M_5 &= p_1 \cdot p_3\end{aligned}$$

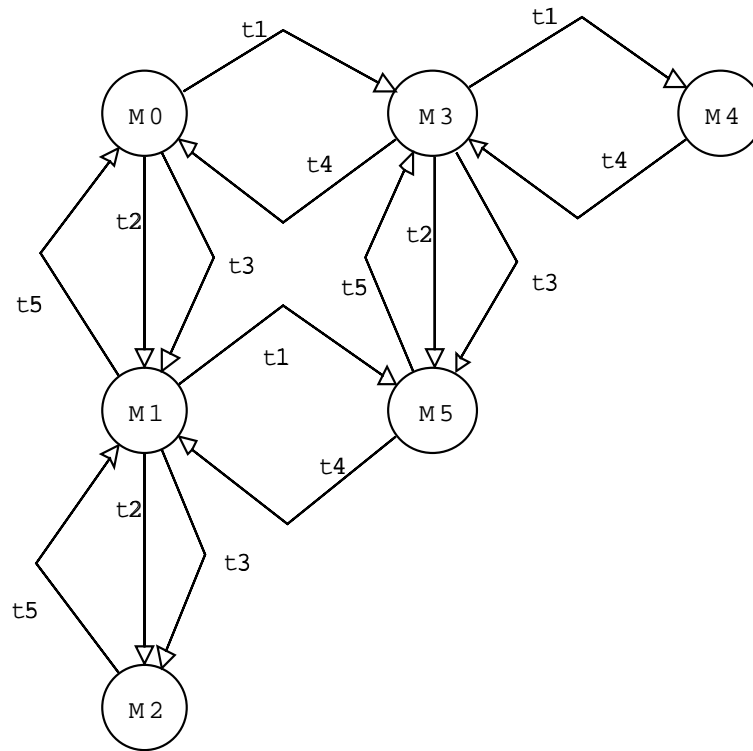


Figure 4.17: Graphe des marquages accessibles.

$$\begin{array}{c}
 M_0 \\
 M_1 \\
 M_2 \\
 M_3 \\
 M_4 \\
 M_5
 \end{array}
 \begin{bmatrix}
 M_0 & M_1 & M_2 & M_3 & M_4 & M_5 \\
 -\lambda_1 - \lambda_2 - \lambda_3 & \lambda_2 + \lambda_3 & 0 & \lambda_1 & 0 & 0 \\
 \lambda_5 & -\lambda_1 - \lambda_2 - \lambda_3 - \lambda_5 & \lambda_2 + \lambda_3 & 0 & 0 & \lambda_1 \\
 0 & 2.\lambda_5 & -2.\lambda_5 & 0 & 0 & 0 \\
 \lambda_4 & 0 & 0 & -\lambda_1 - \lambda_2 - \lambda_3 - \lambda_4 & \lambda_1 & \lambda_2 + \lambda_3 \\
 0 & 0 & 0 & \lambda_4 & -\lambda_4 & 0 \\
 0 & \lambda_4 & 0 & \lambda_5 & 0 & -\lambda_4 - \lambda_5
 \end{bmatrix}$$

Figure 4.18: Matrice des taux de transition.

Remarques:

- La première colonne de la matrice de la figure 4.18 donne l'équation suivante (π_i est la probabilité du marquage M_i):

$$\frac{d\pi_0}{dt} = -(\lambda_1 + \lambda_2 + \lambda_3) \cdot \pi_0 + \lambda_5 \cdot \pi_1 + \lambda_4 \cdot \pi_3$$

- La somme des éléments d'une ligne de Q est toujours égale à zéro car les arcs sortants d'un état M_i sont aussi les arcs aboutissant aux états M_j à partir de M_i .

Table des Matières

1	Un peu d'histoire.	1
2	Définitions.	3
2.1	Concepts de base.	3
2.1.1	Réseau de Petri.	3
2.1.2	Réseau marqué.	3
2.1.3	Graphe associé et notations matricielle.	4
2.1.4	Réseau de Petri pur.	5
2.1.5	Transition franchissable.	5
2.1.6	Franchissement d'une transition.	6
2.1.7	Conflit et parallélisme.	6
2.1.8	Séquence de franchissement.	7
2.1.9	Ensemble des marquages accessibles.	9
2.2	Ensemble de règles et réseau de Petri.	9
2.2.1	Réseau de Petri.	9
2.2.2	Système de règles.	10
2.2.3	Grammaire.	10
2.2.4	Exemple.	11
2.2.5	Conclusion.	12
2.3	"Bonnes" propriétés.	12
2.3.1	K-borné.	12
	a) Place k-bornée et binaire.	12
	b) Réseau de Petri marqué k-borné et binaire.	13
	c) Exemples.	13
	d) Remarques.	13
2.3.2	Vivant.	14
	a) Transition quasi-vivante.	14
	b) Transition vivante.	14
	c) Exemple.	14
	d) Réseau marqué vivant.	16
2.3.3	Réinitialisable.	17
	a) Réseau marqué réinitialisable.	17
	b) Exemple.	17
2.3.4	Remarque finale.	18

2.4	Composantes conservatives et répétitives stationnaires.	18
2.4.1	Composantes conservatives, invariants de place.	18
	Composantes conservatives positives:	20
2.4.2	Composantes répétitives, invariants de transition.	20
	Composantes répétitives stationnaires positives:	21
3	Analyse des Propriétés.	23
3.1	Analyse par énumération des marquages.	23
3.1.1	La propriété de réseau k-borné est décidable.	23
3.1.2	Arbre de couverture.	24
3.1.3	Recherche des propriétés.	25
3.2	Analyse par réduction.	26
3.2.1	Place substituable.	26
3.2.2	Place implicite.	27
	a) Principe:	27
	b) Exemple:	28
	c) Remarques:	29
	d) Places identiques et places implicites dégénérées:	31
3.2.3	Transitions neutres ou identités.	31
3.2.4	Transitions identiques.	33
3.3	Calcul des composantes conservatives et répétitives stationnaires.	33
3.3.1	Principe du calcul d'une base.	33
3.3.2	Exemple.	38
3.3.3	Algorithme simplifié et recherche des solutions positives.	39
3.4	Relation entre les diverses méthodes d'analyse.	41
3.4.1	Obtention de "bonnes" propriétés par le calcul des composantes.	41
	Couverture de place et borné	42
	Borné, vivant et couverture de transition	42
3.4.2	Complémentarité entre l'analyse par réduction et le calcul des composantes.	43
3.4.3	Caractérisation des marquages.	43
4	Les données et le temps.	47
4.1	Introduction.	47
4.2	Interprétation d'un réseau de Petri.	47
4.2.1	Sémantique des places, des transitions et des jetons.	47
4.2.2	Interaction avec les données et l'environnement.	48
4.2.3	Contrôle, données et environnement.	49
4.2.4	Prise en compte explicite du temps.	50
4.3	Retour sur l'analyse.	50
4.3.1	Résultats principaux.	50
4.3.2	Introduction à divers modèles.	51
4.4	Réseau de Petri Colorés.	52
4.4.1	Repliage d'un réseau de Petri.	52

4.4.2	Association de couleurs aux jetons.	52
4.4.3	Association de fonctions aux arcs.	55
4.4.4	Définition formelle (simplifiée).	56
4.4.5	Exemple.	57
4.4.6	Analyse.	58
4.5	Réseaux de Petri Prédicats-Transitions.	59
4.5.1	Notion de variables, semi-unification.	59
	Remarque:	60
4.5.2	Notion de n-uplet de constantes et de variables.	60
4.5.3	Définition (simplifiée) des réseaux Prédicats-Transitions.	61
	Remarques:	62
4.5.4	Exemple.	62
4.5.5	Représentation par réseau de Petri d'un système de règles.	63
4.6	Réseaux de Petri à Objets.	65
4.6.1	Notion d'objet.	65
4.6.2	Définition (simplifiée) des réseaux de Petri à Objets.	66
4.6.3	Exemple.	67
4.6.4	Analyse.	67
4.7	Réseaux de Petri temporisés.	68
4.7.1	Temps associé à une place:	70
4.7.2	Temps associé à une transition:	70
4.7.3	Définition.	70
4.8	Réseaux de Petri temporels.	71
4.8.1	Représentation d'un chien de garde.	71
4.8.2	Définition.	71
4.8.3	Comparaison avec les autres modèles.	72
4.8.4	Analyse	72
4.8.5	Exemple	73
4.9	Réseaux de Petri stochastiques.	75
4.9.1	Limite des réseaux temporisés et temporels pour l'analyse.	75
4.9.2	Durée de sensibilisation stochastique.	75
4.9.3	Définition des réseaux de Petri stochastiques.	76
	Remarque.	76
4.9.4	Obtention d'un processus de Markov.	76
4.9.5	Exemple.	77
	Remarques:	79

Liste d'ouvrages
sur les
Réseaux de Petri
—o—

- G.W. Brams: *Réseaux de Petri: Théorie et Pratique, Tome 1 théorie et analyse*, Editions Masson, 1982, ISBN 2-903-60712-5.
- G.W. Brams: *Réseaux de Petri: Théorie et Pratique, Tome 2 Modélisation et applications*, Masson, 1983, ISBN 2-903-60713-3.
- W. Brauer (ed.): *Net Theory and Applications*, Springer-Verlag Lecture Notes in Computer Science 84 (1980), Proceedings of the advanced course on general net theory of processes and systems, Hamburg.
- W. Brauer (ed.): *Petri Nets: Central Models and Their Properties*, Springer-Verlag Lecture Notes in Computer Science 254 (1987),
- W. Brauer (ed.): *Petri Nets: Applications and Relationships to Other Models of Concurrency* Springer-Verlag Lecture Notes in Computer Science 255 (1987),
- M. Courvoisier, R. Valette: *Commande des Procédés Discontinus - Logique Séquentielle*, Dunod Université, Bordas 1986, ISBN 2-04-016423-5.
- R. David, H. Alla: *Du Grafset aux Réseaux de Petri*, Editions Hermes (2^e édition revue et augmentée), Paris, 1992, ISBN 2-86601-325-5.
- René David, Hassane Alla: *Petri nets and Grafset: tools for modelling discrete event systems*, Prentice Hall International, UK, 1992, ISBN 0-13-327537-X.
- K. Jensen, G. Rozenberg: *High-level Petri nets: Theory and application*, Springer Verlag 1991 ISBN 3-540-54125-X.
- T. Murata: *Petri Nets: Properties, Analysis and Applications*, Proceedings of the IEEE, Vol.77, No 4, p. 541-580 (April 1989).
- J.L. Peterson: *Petri Net Theory and the Modelling of Systems*, Prentice-Hall 1981.
- W. Reisig: *Petrinetze, eine Einführung*, Springer Verlag 1982, ISBN 3-540-11478-5 et ISBN 0-387-11478-5.
- W. Reisig: *Petri Nets: an Introduction*, EATCS Monographs on Theoretical Computer Science, Springer Verlag, Berlin 1985, ISBN 0-387-13723-8.
- W. Reisig: *A primer in Petri net design*, Springer compass, Springer Verlag 1992, ISBN 3-540-13786-6.

- C. Reutenauer: *Aspects Mathématiques des Réseaux de Petri*, Masson, Etudes et Recherches en Informatique, (1989).
- M. Silva: *Las redes de Petri en la Automática y la Informática*, Ed. AC, Madrid (1985).
- G. Vidal-Naquet, A. Choquet-Geniet: *Réseaux de Petri et systèmes parallèles*, Ed. Armand Colin, collection 2ai, 1992, ISBN 2-200-21197-X.