Vérification de la consistance et de la conservation de Petri

Chengbin Chu, Feng Chu, Jean-Marie Proth

INRIA

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

Vérification de la consistance et de la conservation d'un réseau de Pétri

Chengbin CHU
Feng CHU
Jean-Marie PROTH

N° 2070
Octobre 1993

PROGRAMME 5

Traitement du signal,
automatique et
productique

Vérification de la Consistance et de la Conservation des Réseau de Petri (Verification of the Consistency and the Conservativeness of Petri Nets)

Chengbin Chu(*), Feng Chu(*) et Jean-Marie Proth(**)

(*) INRIA-Lorraine, CESCOM, Technopôle Metz 2000, 4 rue Marconi, 57070 Metz, FRANCE.

(**) INRIA-Lorraine, CESCOM, Technopôle Metz 2000, 4 rue Marconi, 57070 Metz, FRANCE et Institute for Systems Research, University of Maryland, College Park, MD 20742, USA.

Résumé. Les réseaux de Petri sont un outil performant pour concevoir, modéliser, analyser et évaluer les systèmes à événements discrets. Dans ce rapport, nous étudions des propriétés structuelles importantes des réseaux de Petri utilisés en gestion de production : la consistance et la conservation. Nous mettons en évidence une condition nécessaire et suffisante simple pour vérifier la consistance et la conservation. Cette condition permet de diminuer la dimension du problème. Nous dégageons des conditions suffisantes dans des cas particuliers. Nous proposons en outre un algorithme pour vérifier ces deux propriétés dans le cas général.

Mots clés. Réseaux de Petri, Consistance, Conservation.

Abstract. Petri Nets are a powerful tool to design, model, analyze and evaluate discrete event systems. This paper addresses the problem of two structural properties which are important in production management: the consistency and the conservativeness. We show a simplified necessary et sufficient condition to verify consistency and conservativeness. This condition allows us to reduce the size of problem. Sufficient conditions are proposed in some particular cases, and an algorithm is developped to verify these two properties in the general case.

Key Words. Petri Nets, Consistency, Conservativeness.

Vérification de la Consistance et de la Conservation des Réseau de Petri (Verification of the Consistency and the Conservativeness of Petri Nets)

Chengbin Chu(*), Feng Chu(*) et Jean-Marie Proth(**)

(*) INRIA-Lorraine, CESCOM, Technopôle Metz 2000, 4 rue Marconi, 57070 Metz, FRANCE.

(**) INRIA-Lorraine, CESCOM, Technopôle Metz 2000, 4 rue Marconi, 57070 Metz, FRANCE et Institute for Systems Research, University of Maryland, College Park, MD 20742, USA.

Résumé. Les réseaux de Petri sont un outil performant pour concevoir, modéliser, analyser et évaluer les systèmes à événements discrets. Dans ce rapport, nous étudions des propriétés structuelles importantes des réseaux de Petri utilisés en gestion de production : la consistance et la conservation. Nous mettons en évidence une condition nécessaire et suffisante simple pour vérifier la consistance et la conservation. Cette condition permet de diminuer la dimension du problème. Nous dégageons des conditions suffisantes dans des cas particuliers. Nous proposons en outre un algorithme pour vérifier ces deux propriétés dans le cas général.

Mots clés. Réseaux de Petri, Consistance, Conservation.

Abstract. Petri Nets are a powerful tool to design, model, analyze and evaluate discrete event systems. This paper addresses the problem of two structural properties which are important in production management: the consistency and the conservativeness. We show a simplified necessary et sufficient condition to verify consistency and conservativeness. This condition allows us to reduce the size of problem. Sufficient conditions are proposed in some particular cases, and an algorithm is developped to verify these two properties in the general case.

Key Words. Petri Nets, Consistency, Conservativeness.

0. Introduction

Les réseaux de Petri sont utilisés de plus en plus fréquemment pour modéliser, analyser et évaluer les systèmes à événements distcrets, et en particulier les systèmes de production (Murata 1989 et Proth 1992). Dans la théorie de réseaux de Petri, les propriétés structuelles et les propriétés comportementales sont deux aspects importants. Les propriétés comportementales ont été étudiées par de nombreux chercheurs (Hillion et Proth 1989 , Laftit et al. 1992, etc). Les propriétés structuelles ont également été étudiées. En utilisant l'équation fondamentale, plusieurs algorithmes ont été développés pour l'analyse des invariants (Silva et Colom 1988). Teruel et Silva (1993) ont étudié des sous-classes de réseaux de Petri appelés systèmes à "conflits égaux". Ils ont démontré que la vivacité d'un tel système borné peut être décidée en vérifiant s'il n'existe pas de solution pour un système d'inégalités linéaires dans le domaine d'intégration et qu'un système à conflits égaux vivant et borné a des états stables.

Dans ce rapport, nous étudions les réseaux de Petri généraux qui modélisent des systèmes de production. Nous nous intéressons à deux propriétés structuelles d'un réseau de Petri : la consistance et la conservation. Nous mettons en évidence une condition nécessaire et suffisante simple pour vérifier la consistance et la conservation. Cette condition permet de reduire la dimension du problème et de vérifier directement la consistance et la conservation dans certains cas particuliers. Comme le montrent les expériences numériques, si un système n'est pas consistant ou conservatif, on tombe souvent dans un des cas particuliers que nous mettons en évidence. Dans le cas général, nous proposons un algorithme pour vérifier la consistance et la conservation d'un réseau de Petri. Les résulats numériques montrent l'efficacité de cet algorithme.

Ce rapport est organisé en 5 sections. La section 1 est un rappel des définitions et des propriétés des réseaux de Petri. La section 2 étudie la consistance et la conservation d'un réseau de Petri. La section 3 présente un algorithme, apellé méthode de Dantzig améliorée, pour vérifier l'existence d'une solution non négative d'un système d'équations linéaires. La section 4 donne des résultats numériques. La section 5 conclut ce rapport.

1. Rappels sur les réseaux de Petri

Dans cette section, nous rappelons quelques notions fondamentales et des définitions de propriétés importantes dans la théorie des réseaux de Petri.

Un **réseau de Petri** est un graphe bi-partie dont les noeuds sont des places et des transitions. Les arcs relient des places à des transitions et des transitions à des places. Un réseau de Petri est souvent représenté par un quadruplet N = (P,T,A,W), où:

$P = \{p_1, p_2, ..., p_m\}$ est l'ensemble des places,

$T = \{t_1, t_2, ..., t_n\}$ est l'ensemble des transitions,

A est l'ensemble des arcs,

W est la valuation des arcs qui est une fonction à valeurs positives entières.

Si (t, p) (resp. (p, t)) $\in$ A, on dira que p est une place de sortie (resp. d'entrée) de la transition t et que t est une transition d'entrée (resp. de sortie) de la place p. On désigne par $t^\bullet$ (resp. $^\bullet t$) l'ensemble des places de sortie (resp. d'entrée) de la transition t, et par $p^\bullet$ (resp. $^\bullet p$) l'ensemble des transitions de sortie (resp. d'entrée) de la place p.

Une transition t est dite **franchissable** si et seulement si le nombre de jetons dans chacune de ses places d'entrée $p \in {}^\bullet t$ est supérieur ou égal à la valuation de l'arc (p, t).

Si une séquence de transitions σ est franchissable pour un marquage M_0, alors nous arrivons à un autre marquage M après le tir des transitions de σ, et nous écrivons $M_0 \xrightarrow{\sigma} M$.

Dans ce rapport, nous supposons que les réseaux de Petri sont purs, c'est-à-dire qu'il n'y a pas de transitions de réentrance. Autrement dit ${}^\bullet P \cap P^\bullet = \varnothing$. En effet les réseaux de Petri impurs peuvent toujours être transformés en réseaux de Petri purs.

$C = [c_{i,j}]$ est la matrice d'incidence, i = 1, 2, ..., m et j = 1, 2, ..., n,

$$c_{i,j} = \begin{cases} +w(t_j, p_i) & \text{si } p_i \in t_j^\bullet \\ -w(p_i, t_j) & \text{si } t_j \in p_i^\bullet \\ 0 & \text{sinon} \end{cases}$$

On désigne par $\overline{\sigma}$ le **vecteur caractéristique** de la séquence σ. Sa j-ième composante représente le nombre d'occurrences de la transition t_j dans σ.

Alors l'équation fondamentale s'écrit: $M = M_0 + C\overline{\sigma}$.

Elle donne le marquage M obtenu à partir de M_0 si l'on franchit les transitions de σ, en supposant que σ soit franchissable.

Un vecteur entier colonne $X \geq 0$ de dimension n est un **t-invariant** si et seulement si $CX = 0$.

Un réseau de Petri est **consistant** si il existe un marquage M_0 et une séquence σ franchissable tels que $M_0 \xrightarrow{\sigma} M_0$ et que chaque transition figure au moins une fois dans σ. On démontre qu'un réseau de Petri est consistant (resp. partiellement consistant) si et seulement si il existe un vecteur X strictement positif (resp. non négatif) tel que $CX = 0$.

Un vecteur entier colonne $Y \geq 0$ de dimension m est un **p-invariant** (appelé aussi p-semi-flot) si et seulement si $M^tY = M_0^tY$ (i.e. $C^tY = 0$) pour un marquage initial quelconque M_0 et pour un marquage atteignable quelconque M, où M^t désigne la transposée de la matrice M.

Un réseau de Petri est **conservatif** si il existe un vecteur colonne Y de dimension m, strictement positif tel que $M^tY = M_0^tY$ est une constante pour chaque marquage atteignable M et pour un marquage initial quelconque M_0. On démontre qu'un réseau de Petri est conservatif si et seulement si il existe un vecteur colonne Y, de dimension m strictement positif tel que $C^tY = 0$.

Nous allons montrer que la vérification de la consistance et de la conservation peut se faire à l'aide du même algorithme qui calcule si il existe une solution strictement positive pour l'équation $AX = 0$. Si l'on veut vérifier la consistance (resp. conservation) il suffit de remplacer A par C (resp. C^t).

2. Vérification de l'existence d'une solution

Dans cette section, nous montrons comment vérifier l'existence d'au moins une solution strictement positive au système d'équations $AX = 0$. Cette vérification nous permet de vérifier

la consistance et la conservation d'un réseau de Petri. Avant de commencer la vérification proprement dite qui va être détaillée dans la section suivante, nous cherchons à réduire la dimension de ce système et à examiner des cas particulieirs.

Propriété 1 (Mehta 1989):

Pour une matrice quelconque A de dimension m × n, dont le rang est r (c'est-à-dire rang(A) = r), il existe deux matrices D et E, respectivement de dimension m × r et r × n, telles que A = D×E, et rang(D) = rang(E) = r.

Les matrices D et E ne sont pas uniques. Une méthode pour calculer ces matrices est la suivante, où $\hat{A}_{i\bullet}$ (resp. $\hat{A}_{\bullet i}$) est le vecteur ligne (resp. colonne) constitué des éléments de la i-ième ligne (resp. colonne) de la matrice $\hat{A}$.

Algorithme 1:

1. $\hat{A}$=A, i=1;
2. Chercher la première ligne notée i* comportant au moins un élément non nul;
3. **Si** i* n'existe pas, **alors** on s'arrête;
4. **Si** i*≠i, **alors** ajouter la ligne i* à la ligne i;
5. Chercher le premier élément $\hat{a}_{i,j*}$ non nul;
6. Diviser la ligne i par $\hat{a}_{i,j*}$;
7. Ajouter à toute ligne i' autre que la ligne i une ligne $-\hat{a}_{i',j*} \times \hat{A}_{i\bullet}$. Ceci annule les éléments de la colonne j* sauf celui de la ligne i.
8. i=i+1, **aller à** 2.

En utilisant cet algorithme, on obtient une matrice $\hat{A}$, de la forme suivante (où * signifie que l'élément peut être non nul):

$$\hat{A} = \begin{pmatrix} 0\cdots0 & 1 & *\cdots* & 0 & *\cdots* & 0 & *\cdots* \\ 0\cdots0 & 0 & 0\cdots0 & 1 & *\cdots* & 0 & *\cdots* \\ \cdots\cdots & \cdots & \cdots\cdots & \cdots & \cdots\cdots & \cdots & \cdots\cdots \\ 0\cdots0 & 0 & 0\cdots0 & 0 & 0\cdots0 & 1 & *\cdots* \\ 0\cdots0 & 0 & 0\cdots0 & 0 & 0\cdots0 & 0 & 0\cdots0 \\ \cdots\cdots & \cdots & \cdots\cdots & \cdots & \cdots\cdots & \cdots & \cdots\cdots \\ 0\cdots0 & 0 & 0\cdots0 & 0 & 0\cdots0 & 0 & 0\cdots0 \end{pmatrix} \begin{matrix} \\ \\ \\ \\ \\ \\ \end{matrix} \begin{matrix} \} r \\ \\ \\ \\ \} n-r \\ \\ \end{matrix}$$

$$\qquad\qquad (k_1) \qquad (k_2) \qquad (k_r)$$

Les colonnes $k_1, k_2, \ldots, k_r$ de $\hat{A}$ sont indépendantes et leur restriction aux r premières lignes (i.e. à la matrice E) constitue une matrice unité.

On montre qu'une solution consiste à prendre les colonnes $k_1, k_2, \ldots, k_r$ de la matrice A pour construire D, c'est-à-dire,

$$D = (A_{\bullet k_1}, A_{\bullet k_2}, \ldots, A_{\bullet k_r}),$$

et à choisir les r premières lignes de $\hat{A}$, comme matrice E, c'est -à-dire que

$$E = \begin{pmatrix} 0\cdots0 & 1 & *\ldots* & 0 & *\ldots* & 0 & *\ldots* \\ 0\cdots0 & 0 & *\ldots* & 1 & *\ldots* & 0 & *\ldots* \\ \ldots\ldots & \ldots & \ldots\ldots & \ldots & \ldots\ldots & \ldots & \ldots\ldots \\ 0\cdots0 & 0 & 0\cdots0 & 0 & 0\cdots0 & 1 & *\ldots* \end{pmatrix}.$$

Propriété 2:

Une condition nécessaire et suffisante pour qu'un vecteur X soit la solution de AX = 0 est que

$$EX = 0.$$

Démonstration:

Il est évident que c'est une condition suffisante, en effet,

$$EX = 0 \quad \Rightarrow \quad DEX = 0 \quad \Rightarrow \quad AX = 0.$$

On démontre maintenant que c'est également une condition nécessaire.

$$AX = 0 \quad \Rightarrow \quad DEX = 0.$$

Comme rang(D) = r, il existe r lignes independantes. On peut permuter les lignes pour faire apparaître D_1 et D_2 comme sous-matrices de D, respectivement de dimensions $r \times r$ et $(m\text{-}r) \times r$, telles que rang $(D_1) = r$.

Alors $$DEX = 0 \quad \Rightarrow \quad \begin{bmatrix} D_1 \\ D_2 \end{bmatrix} EX = 0 \quad \Rightarrow \quad D_1EX = 0$$

La matrice D_1 n'est pas singulière, parce que c'est une matrice carrée de dimension $r \times r$ et $\text{rang}(D_1) = r$, alors $(D_1)^{-1}$ existe et

$$EX = (D_1)^{-1}[0] \qquad \Rightarrow \qquad EX = 0.$$

Q.E.D.

Ainsi, si rang(C)=r, on peut trouver deux matrices D et E respectivement de dimension $m \times r$ et $r \times n$ et telles que $C = D \times E$ et que rang(D)=rang(E)=r. On peut vérifier la consistance en vérifiant s'il existe une solution striuctement positive au système EX=0.

Comme rang (E) = r, il existe r colonnes indépendantes. Après avoir permuté les colonnes, on peut obtenir E_1 et E_2 comme sous-matrices de E, respectivement de dimension $r \times r$ et $r \times (n-r)$, telles que rang $(E_1) = r$. Dans ce cas, E_1 est inversible.

Alors la vérification de la consistance consiste à vérifier s'il existe deux vecteurs X_1 et X_2 strictement positifs respectivement de dimension r et n-r tels que

$$[E_1 \quad E_2]\begin{bmatrix} X_1 \\ X_2 \end{bmatrix} = 0$$

i.e. $E_1X_1 + E_2X_2 = 0,$

ou encore $X_1 = -(E_1)^{-1}E_2X_2 .$

Si la matrice E est obtenue avec la méthode présentée plus haut, E_1 peut être une matrice identité, en prenant les colonnes $k_1, k_2, ..., k_r$. Par conséquent la matrice $(E_1)^{-1}$ est également une matrice identité. Les autres colonnes de E constituent E_2. Ainsi, on a

$$X_1 = -E_2X_2 .$$

Comme $X_1 > 0$ et $X_2 > 0$, on obtient la propriété suivante.

Propriété 3:

La condition nécessaire et suffisante pour qu'un système soit consistant est qu'il existe un vecteur X_2 positif, tel que

$$E_2X_2 < 0. \qquad (1)$$

D'une manière similaire, on peut trouver deux matrices F et G respectivement de dimension $n \times r$ et $r \times m$ et telles que $C^t = F \times G$ et que rang(F)=rang(G)=r. On peut ainsi vérifier la conservation en vérifiant l'existence d'une solution strictement positive au système GY=0.

Comme rang (G) = r, il existe r lignes indépendantes. Après avoir permuté les colonnes, on peut avoir G_1 et G_2 comme sous-matrices de G, respectivement de dimension $r \times r$ et $r \times (m-r)$, telles que rang $(G_1) = r$. G_1 est alors inversible. La vérification de la conservation consiste donc à vérifier s'il existe deux vecteurs Y_1 et Y_2 strictement positifs respectivement de dimension r et m-r tels que

$$[G_1 \quad G_2]\begin{bmatrix} Y_1 \\ Y_2 \end{bmatrix} = 0$$

i.e. $G_1Y_1 + G_2Y_2 = 0,$

ou encore $Y_1 = -(G_1)^{-1}G_2Y_2 .$

Si la matrice G est obtenue avec l'algorithme 1, G_1 est une matrice identité Par conséquent, on a

$$Y_1 = -G_2Y_2 .$$

Comme $Y_1 > 0$ et $Y_2 > 0$, on obtient la propriété suivante.

Propriété 4:

La condition nécessaire et suffisante pour qu'un système soit conservatif est qu'il existe un vecteur Y_2 positif, tel que

$$G_2Y_2 < 0. \qquad (2)$$

Si la matrice E_2 (resp. G_2) vérifie un des trois cas particuliers souvants, on peut alors vérifier immédiatement la consistance (resp. conservation).

1) Si les éléments d'une colonne de E_2 (resp. G_2) sont tous négatifs, alors le système (1) (resp. (2)) a une solution et le système est consistant (resp. conservatif). En effet, il suffit de prendre l'élément de X_2 (resp. Y_2)correspondant à cette colonne suffisamment grand, et les autres éléments de X_2 (resp. Y_2) suffisamment petits.

2) Si la somme des éléments de chaque ligne de E_2 (resp. G_2) est négative, alors le système est consistant (resp. conservatif) (il suffit de prendre tous les x (resp. y) à 1).

3) Si une ligne de E_2 (resp. G_2) ne comporte que des éléments positifs ou nuls, alors le système n'est pas consistant (resp. conservatif).

Dans le cas général, nous faisons une transformation équivalente de manière à pouvoir utiliser la méthode de Dantzig. En effet, les trois énoncés suivants sont équivalents

1. Le système AX=0, X>0 a une solution,
2. Le système AX=0, X≥e a une solution,
3. Le système AX = -Ae, X≥0 a une solution,

où e est un vecteur unité.

Le troisième énoncé peut être résolu par la méthode de Dantzig améliorée présentée dans la section suivante. Par conséquent, la vérification de la consistance et de la conservation peuvent s'effectuer en remplaçant A respectivement par E et G.

3. Méthode de Dantzig améliorée

La méthode de Dantzig permet de vérifier s'il existe une solution à l'équation AX=B, telle que X≥0, où A est une matrice de dimension p×q, B est un vecteur colonne de dimension p, X est un vecteur colonne de dimension q. On suppose que le rang de A est p (Kaufmann 1970).

On choisit arbitrairement une base formée de p-1 vecteurs P_i (i = 1, ..., p-1) et P_0 = B de telle sorte que la matrice $P=[P_0, P_1, \ldots, P_{p-1}]$ soit inversible. Il faut que dans notre cas, A = E ou G, p = r. Il est toujours possible de trouver de tels vecteurs P_i ($i \geq 1$) pour que $[P_0,P_1,\ldots, P_{p-1}]$ soit inversible. A moins que B = 0, au quel cas, la solution est immédiate : il suffit de prendre tous les x à 0. Si B ≠ 0, alors $b_{i^*} \neq 0$, alors il suffit de prendre $P_i = A_{\bullet k_i}\ \forall i < i^*$, $P_i = A_{\bullet(k_{i+1})}\ \forall i \geq i^*$.

Ainsi, chacun des autres vecteurs peut être exprimé comme suit:

$$P_j = z_{0,j}P_0 + z_{1,j}P_1 + \cdots + z_{r-1,j}P_{r-1} = \sum_{i=0}^{p-1} z_{i,j}P_i .$$

Voici l'algorithme qui vérifie l'existence d'une solution.

Algorithme 2:

1° Déterminer les matrices P et $\overline{P}$ qui est constituée des colonnes de A autres que $P_1, P_2,\ldots, P_{p-1}$.

2° Choisir arbitrairement p-1 coefficients positifs w_i pour les vecteurs faisant partie de la base.

3° Calculer

$$Z = P^{-1}\overline{P}.$$

4° Vérifier $z_{0,j}$.

Si $z_{0,j} \leq 0$ pour tout j = 1,2, ..., q-p+1. Alors il n'y a pas de solution possible

Sinon, *sélectionner un* $j \geq 1$ *tel que* $z_{0,j}$ *soit le plus grand. Ce qui donne le vecteur* P_j *qui va entrer dans la base.*

Si $z_{i,j} \leq 0$ pour tout i = 1, ..., p. Alors il existe une solution.

Sinon, *nous choisissons le vecteur de base qui va être remplacé par* P_j.

On considère une quantité

$$\theta = \min_i(w_i / z_{i,j}) \qquad \text{pour les valeurs de i telles que } z_{i,j} > 0 \text{ et } i > 0.$$

On obtient ainsi le vecteur P_i à retirer de la base : la colonne de $\overline{P}$ numéroté j et la colonne de P numéroté i se permutent. Après les avoir permutées, on obtient un nouveau P^1 et $\overline{P}$.

5° Chercher les nouveaux coefficients.

$$w_i = \theta;$$
$$w_k = w_k - \theta z_{k,j} \qquad k = 1,2, i-1, i+1, \ldots, p-1.$$

6° Recommencer en 3°.

Après un nombre fini d'itérations, on peut toujours savoir s'il existe une solution.

Considérons un exemple de stockeur de type LIFO représenté dans la figure 1 dont le modèle de réseau de Petri est illustré dans la figure 2. Ce type de stockeur est souvent utilisé dans des systèmes de production, surtout dans la fabrication de verre.

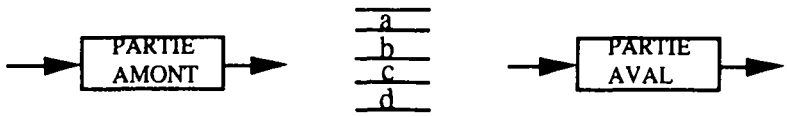

Fig. 1 Stockeur LIFO

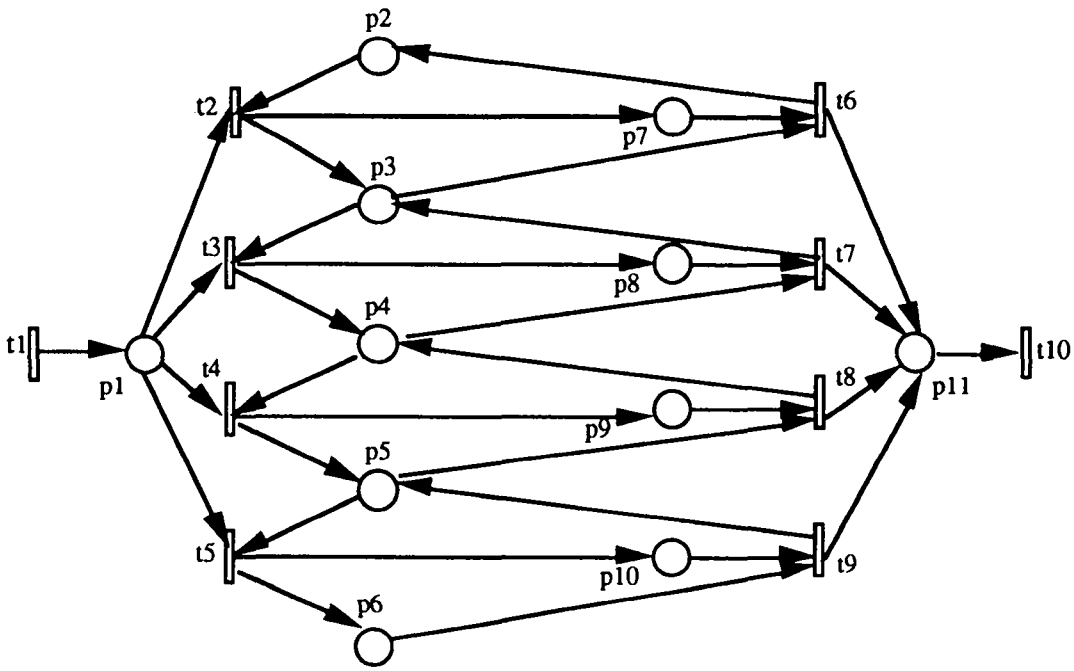

Fig.2 Modèle de réseau de Petri d'un stockeur LIFO

La matice d'incidence C de ce modèle est:

$$C = \begin{pmatrix} 1 & -1 & -1 & -1 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 & -1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 & 0 & -1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & -1 & 0 & 0 & -1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & -1 \end{pmatrix}$$

Après utilisation de l'algorithme 1 pour C, on obtient les matrices E et E_2 suivantes:

$$E = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & -1 \\ 0 & 0 & 1 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & -1 \end{pmatrix} \qquad E_2 = \begin{pmatrix} 0 & 0 & 0 & -1 \\ 1 & 1 & 1 & -1 \\ -1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 1 & 1 & 1 & -1 \end{pmatrix}$$

La matrice E_2 ne correspond pas à un cas particulier. On est obligé d'utiliser l'algorithme 2. On détermine que ce système est consistant en 0,083 seconde de CPU sur un IBM RISC 6000.

pour vérifier la conservation en utilisant de l'algorithme 1 pour C^T, on obtient les matrices G et G_2 suivantes:

$$G = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & -1 & -1 & -1 & -1 & -1 & 0 \\ 0 & 0 & 1 & 0 & 0 & -1 & 0 & -1 & -1 & -1 & 0 \\ 0 & 0 & 0 & 1 & 0 & -1 & 0 & 0 & -1 & -1 & 0 \\ 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \qquad G_2 = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ -1 & -1 & -1 & -1 & -1 \\ -1 & 0 & -1 & -1 & -1 \\ -1 & 0 & 0 & -1 & -1 \\ -1 & 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

La matrice G correspond au cas particulier (3). On sait donc que ce système n'est pas conservatif. Ce calcul a coûté 0,07 seconde de CPU sur un IBM RISC 6000. Ce modèle est un réseau de Petri avec des transitions d'entrée et des transitions de sortie. Ce qui explique qu'il ne soit pas conservatif.

4. Résultats numériques

Pour évaluer les performances de l'algorithme, nous générons des exemples au hasard en nous donnant le nombre de places et le nombre de transitions d'un réseau de Petri sur lequel nous appliquons cet algorithme. Nous choisissons des réseaux de Petri généraux mais à valuation unitaire. Une place comporte au plus 5 transitions d'entrée et 5 transitions de sortie. Utilisant cet algorithme, on peut vérifier la consistance et la conservation d'un réseau de Petri. Les expériences ont été effectués sur un IBM RISC 6000.

Le tableau suivant présente les résultats numériques sur la consistance et la conservation des réseaux de Petri. A partir de ce tableau, on peut savoir si un réseau de Petri est consistant et conservatif et on peut comparer le temps de calcul pour vérifier la consistance et la conservation pour des réseaux de Petri de différente taille, où:

n° est le numérode l'exemple;

NP est le nombre de places;

NT est le nombre de transitions;

V_1 (resp. V_2) est le résultat de la vérification de la consistance (resp. conservation).(C et N signifient respectivement consistant (resp. conservatif) et non consistant (resp.non conservatif);

TC_1 (resp. TC_2) est le temps de calcul pour vérifier la consistance (resp. la conservation) (en unités CPU).

Table 1. Vérification de la consistance et de la conservation

n°	NP	NT	V_1	TC_2	V_2	TC_2
1	9	5	C	0,04	C	0,03
2	47	58	N	0,55	N	0,67
3	30	58	C	0,31	N	0,44
4	100	40	N	0,49	C	1,57
5	100	50	N	0,62	N	1,96
6	100	199	C	9,39	N	5,48
7	200	80	N	2,19	C	9,94
8	200	200	N	9,14	N	14,13
9	150	200	C	16,23	N	9,48
10	200	100	N	3,31	C	16,21
11	300	300	N	29,85	N	41,97
12	150	296	C	35,36	N	14,94
13	300	100	N	4,79	C	21,55
14	250	300	C	68,29	N	31,33
15	300	120	N	6,38	C	30,30
16	400	400	N	65,18	N	86,51
17	300	400	C	132,16	N	54,97
18	400	150	N	12,17	C	62,08
19	200	400	C	82,25	N	30,02
20	400	250	N	28,05	N	146,84
21	500	500	N	118,14	N	157,47
22	500	250	N	34,45	C	237,89
23	400	500	C	297,26	N	106,07

24	250	495	C	159,21	N	48,74
25	500	200	N	23,37	C	145,48
26	500	600	C	1163,28	N	176,23
27	600	400	N	93,22	N	373,23
28	600	250	N	40,03	C	248,16
29	400	599	C	401,58	N	122,09
30	550	200	N	25,08	C	138,70
31	600	700	C	882,38	N	280,79
32	500	700	C	781,88	N	206,27
33	350	696	C	414,71	N	114,80
34	700	300	N	64,24	C	458,13
35	700	250	N	46,80	C	290,29
36	700	800	C	1337,65	N	430,43
37	600	800	C	1337,00	N	320,81
38	800	600	N	263,48	N	906,71
39	800	350	N	96,71	C	754,77
40	450	799	C	2174,45	N	197,40

A partir de ce tableau, on peut constater que le temps de calcul augmente avec la dimension d'un réseau, mais en général, cette augmentation est raisonable par rapport à l'augmentation de la taille du système. Donc on peut utiliser cette méthode pour vérifier la consistance et la conservation d'un système de production de taille moyenne. Cette méthode peut s'appliquer à un système quelconque qui peut être modélisé par un réseau de Petri, même à valuation non unitaire.

Il faut noter que quand le nombre de places d'un système est supérieur ou égal au nombre de transitons, c'est -à-dire que $NP \geq NT$. Le système est souvent non consistant. Dans ce ces, la non consistance est souvent identifiée par le cas particulier (3). Dans le cas où $NP \leq NT$, le système est souvent non conservatif. Ceci est également souvent identifiée par le cas particulier (3).

5. Conclusion

Dans ce rapport, nous avons proposé un algorithme pour vérifier la consistance et la conservation d'un réseau de Petri. Les résultats numériques ont montré que cet algorithme est très efficace. A l'occasion de ce travail, nous avons décidé de nous intéresser à la recherche les causes de la non consistance et/ou de la non conservation d'un système de production, et de trouver les moyens pour éviter l'absence de ces propriétés.

Références

[1] J.M. Colom, M. Silva, "Improving the Linearly Based Characterization of P/Tnets", In Advances in Petri nets 90, LNCS 483:113 - 145, Springer-Verlag.

[2] H.P. Hillion, J.M. Proth, "Performance evaluation of job-shop systems using timed event graphs", IEEE transactions on Automatic Control, Vol. '", No. 1, January 1989.

[3] A. Kaufmann, "Méthodes et Modèles de la Recherche Opérationnelle-1 (Les Mathémathiques de l'Entreprise) ", DUNOD, Paris, 1970.

[4] S. Laftit, J.M. Proth, X.L. Xie, "Optimization of invariant criteria for event graphs", IEEE transactions on Automatic Control, Vol. 37, No. 5, pp. 547-555, May 1992.

[5] M.L. Mehta, "Matrix Theory (Selected Topics and Useful Results)", Les Editions de Phisique, Les Ulis, 1989.

[6] Murata T., "Petri Nets: Properties, Analysis and Applications", Proceedings IEEE, Vol. 77, No. 4, April 1989.

[7] J.M. Proth, "Conception et Gestion des Systèmes de Production", Presses Universitaires de France, 1992.

[8] M. Silva, J.M. Colom, " On the Computation of Structural Synchronic Invariants in P/T nets", In Advances in Petri nets 88, LNCS 340: 387 - 417, Springer-Verlag.

[9] E. Teruel, M. Silva, " Liveness and Home States in Equal Conflit Systems", In Advances in Petri nets 93, LNCS 691: 415 - 432, Springer-Verlag.

Unité de Recherche INRIA Lorraine
Technopôle de Nancy-Brabois - Campus Scientifique
615, rue du Jardin Botanique - B.P. 101 - 54602 VILLERS LES NANCY Cedex (France)
Antenne de Metz
Technopôle de Metz 2000 - Cescom - 4, rue Marconi - 57070 METZ (France)

Unité de Recherche INRIA Rennes IRISA, Campus Universitaire de Beaulieu 35042 RENNES Cedex (France)
Unité de Recherche INRIA Rhône-Alpes 46, avenue Félix Viallet - 38031 GRENOBLE Cedex (France)
Unité de Recherche INRIA Rocquencourt Domaine de Voluceau - Rocquencourt - B.P. 105 - 78153 LE CHESNAY Cedex (France)
Unité de Recherche INRIA Sophia Antipolis 2004, route des Lucioles - B.P. 93 - 06902 SOPHIA ANTIPOLIS Cedex (France)

EDITEUR
INRIA - Domaine de Voluceau - Rocquencourt - B.P. 105 - 78153 LE CHESNAY Cedex (France)

ISSN 0249 - 6399