Université Bordeaux I
U.F.R. Mathématiques et Informatique

Master Informatique 2^{eme} année
Année 2003-2004

Mardi 27 janvier 2004, 16 heures - 17 heures 30
Tous documents autorisés.

Formalismes de Description de Systèmes
Alain Griffault

SUJET + CORRIGE

Avertissement

Les deux parties sont indépendantes. Dans la seconde partie, les questions 2.1, ..., 2.10 sont indépendantes mais *progressives* ; et les questions 2.11, 2.12, 2.13 sont indépendantes.

1 Modélisation de systèmes

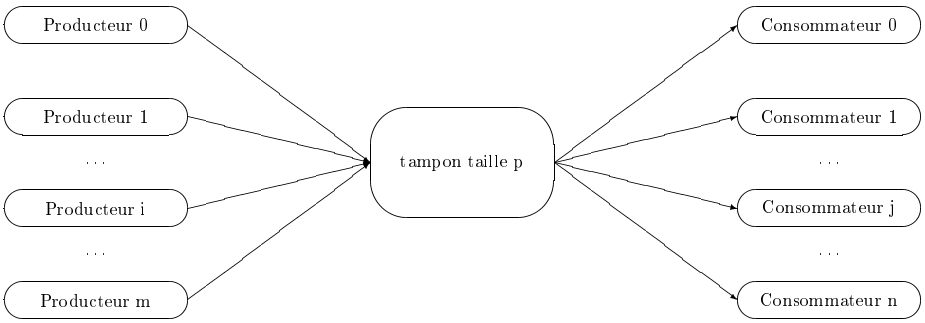
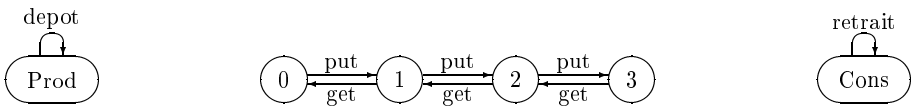


FIG. 1 – Producteurs-Consommateurs avec un tampon borné

Question 1.1 (5 points) *Donnez une description du système de la figure 1 dans un des formalismes vu en cours (systèmes de transition, automates à contrainte, réseau de Petri, Esterel, Electre, Promela, CSP, CCS).*

Produit de systèmes de transition



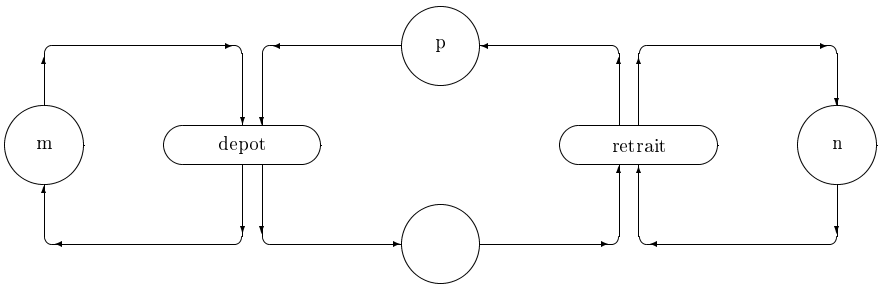
Prod	Prod	Prod	Tampon	Cons	Cons
depot			put		
	depot		put		
		depot	put		
			get	retrait	
			get		retrait

Automates à contraintes

```
Variable nbProd, taille, nbCons, nbMes : entier positif;
event depot, retrait;
transitions
  nbProd > 0 & nbMes < taille |- depot   -> nbMes++;
```

```
  nbCons > 0 & nbMes > 0      |- retrait -> nbMes--;
initial
  nbProd := m;
  taille := p;
  nbCons := n;
  nbMes  := 0;
```

Réseau de Petri



Concurrent Sequential Processes

```
[  Prod1::
     *[true -> Tampon!m;]
// Prod2::
     *[true -> Tampon!m;]
// Tampon::
     nb := 0;
     *[   nb<p & Prod1?m -> nb++;
       [] nb<p & Prod2?m -> nb++;
       [] nb>0 & Cons1?r -> Cons1!m; nb--;
       [] nb>0 & Cons2?r -> Cons2!m; nb--;
      ]
// Cons1::
     *[true -> Tampon!r; Tampon?m;]
// Cons2::
     *[true -> Tampon!r; Tampon?m;]
]
```

Question 1.2 (1 point) *Expliquez vos abstractions et justifiez votre choix de formalisme.*

Pour tous les modèles proposés, tous les messages sont identiques (la valeur du message est considérée comme sans importance).

Produit de systèmes de transition Les modèles pour le producteur et le consommateur sont simples. Le modèle du tampon doit fixé sa taille (ici 3). Le produit doit fixer le nombre de producteur (ici 3), de tampon (ici 1) et de consommateur (ici 2). L'abstraction est classique, il faut ensuite convaincre de sa *validité*. En général, cela est relatif aux propriétés que l'on souhaite vérifiée.

Automates à contraintes Les producteurs et les consommateurs sont abstraits par des constantes (des variables initialisées au nombre de producteurs et au nombre de comsommateurs). Le tampon est abstrait par une constante (une variable initialisée à sa taille) et par une variable qui mémorise le nombre de messages présents. Les transitions sont alors simples à écrire.

Réseau de Petri L'abstraction est la même que la précédente. Pour ce type d'abstraction (plusieurs exemplaires de processus), le réseau de Petri est certainement le formalisme le mieux adapté.

Concurrent Sequential Processes Comme pour le produit de système de transition, CSP impose de fixer le nombre de producteurs et de consommateurs. La taille du tampon est gérée par une variable du processus Tampon.

2 Transformations de modèles

L'objectif de cet exercice est de montrer les liens entre les systèmes de transition à contrainte et les réseaux de Petri.

Définition 1 (Un système de transition à contrainte) *est un triplet :*
- *un ensemble de variables $\vec{s}$ sur des domaines D_{s_i},*
- *un ensemble d'événements E,*
- *un ensemble de transitions :*

$$G(\vec{s}) \quad \xrightarrow{e} \quad \vec{s} := \sigma(\vec{s})$$

$G(\vec{s})$ est un(e) garde (c-à-d une expression booléenne) et $e \in E$,

Un événement e est tirable si la garde associée est vraie, et si le franchissement est possible. Le franchissement consiste à effectuer les affectations décrite dans $\sigma(\vec{s})$. Le franchissement est impossible ssi une des variables reçoit une valeur en dehors de son domaine.

Définition 2 (Un réseau de Petri) *est composé de :*
- *un ensemble de places $P = \{p_1, p_2, \ldots, p_m\}$* **fini**
- *un ensemble de transition $T = \{t_1, t_2, \ldots, t_n\}$* **fini**
- *deux applications*

$$\begin{aligned} Pre: &\quad P \times T \to \quad \{0,1\} \\ Post: &\quad P \times T \to \quad \{0,1\} \end{aligned}$$

  $Pre(p_i, t_j) = 1$ ssi P_i est une place en entrée de T_j ; et $Post(p_i, t_j) = 1$ ssi P_i est une place en sortie de T_j.
- *un marquage initial : une application $M_0 : P \to N$ qui représente le nombre de jetons présents dans chaque place initialement.*

Une transition t_i est franchissable ssi chacune des places en entrée comporte au moins un jeton. Le franchissement consiste à décrémenter d'une unité toutes les places en entrée et à incrémenter d'une unité toutes les places en sortie.

Propriété 1 *Les matrices d'incidences* $W^- = \begin{pmatrix} & \vdots & \\ \ldots & Pre(p_i,t_j) & \ldots \\ & \vdots & \end{pmatrix}$ *et* $W^+ = \begin{pmatrix} & \vdots & \\ \ldots & Post(p_i,t_j) & \ldots \\ & \vdots & \end{pmatrix}$ *et une matrice $M_0 = (p_{1_0}, \ldots, p_{m_0})$ représentent exactement un réseau de Petri et son marquage initial.*

Définition 3 (Un réseau de Petri pur) *est un réseau de Petri pour lequel* $\forall p_i, \forall t_j, (Pre(p_i, t_j) = 1 \& Post(p_i, t_j) = 1) = false$

Question 2.1 (2 points) *En utilisant la notation de votre choix (algorithme, programme C, programme Java), décrivez la fonctionnalité prenant en entrée un* **réseau de Petri pur** *donné sous la forme du triplet de matrice (W^-, W^+, M_0), et fournissant en sortie un systèmes de transition à contrainte (liste des variables, liste des événements, liste des transitions, liste des valeurs initiales) équivalent.*

1. À chaque place p_i, on associe une variable entière $p_i : N$, initialisée à $M_0(p_i)$.
2. À chaque transition t_j, on associe :
   - un événement t_j,
   - une transition $True \xrightarrow{t_j} \left\{ \begin{array}{l} \ldots \\ p_i := p_i - Pre(p_i, t_j) + Post(p_i, t_j) \\ \ldots \end{array} \right.$

La garde $True$ est suffisante car dans les automates à contraintes (déf 1), la postcondition interdit qu'une variable soit affectée en dehors de son domaine. Ici les variables n'ont donc pas le droit de devenir négative.

Définition 4 (Un système de transition à contrainte très simple) *est un triplet :*
- *un ensemble de variables $\vec{s}$ sur N,*
- *un ensemble d'événements E,*
- *un ensemble de transitions :*

$$True \quad \xrightarrow{e} \quad \vec{s} := \sigma(\vec{s})$$

où $\sigma(\vec{s})$ est restreint aux opérations $\{s_i := s_i + 1, s_i := s_i, s_i := s_i - 1\}$.

Question 2.2 (1 point) *Soit un système de transition à contrainte très simple (déf 4), donnez les applications Pre et Post d'un réseau de Petri pur équivalent.*

C'est la transformation inverse de la question précédente.

1. À chaque variable $s_i : N$, on associe une place s_i.
2. À chaque couple (événement, transition) e_j, on associe une transition e_j.
3. Les applications Pre et $Post$ sont construites en traduisant les posconditions :

$$\forall e_j \begin{cases} s_i := s_i - 1 & \Rightarrow & Pre(s_i, e_j) = 1, Post(s_i, e_j) = 0 \\ s_i := s_i & \Rightarrow & Pre(s_i, e_j) = 0, Post(s_i, e_j) = 0 \\ s_i := s_i + 1 & \Rightarrow & Pre(s_i, e_j) = 0, Post(s_i, e_j) = 1 \end{cases}$$

4. $\forall s_i, M_0(s_i) = init(s_i)$.

Question 2.3 (1 point) *Indiquez les modifications (si nécessaire) à apporter à votre solution à la question 2.1, afin d'avoir en entrée un* **réseau de Petri** *donné sous la forme du triplet de matrice (W^-, W^+, M_0), et toujours en sortie un système de transition à contrainte équivalent.*

Le problème provient des transitions *impures*, qui possèdent une place qui est à la fois en entrée et en sortie ($Pre(p_i, t_j) = 1 \& Post(p_i, t_j) = 1$). Le nombre de jetons est inchangée par la transition, mais la place doit contenir au moins un jeton pour que la transition soit franchissable. La garde de la transition t_j devient donc ((négation impure) ou (1 jeton)) pour chaque place.

$$\bigwedge_{i=1}^{i=m} ((Pre(p_i, t_j) = 0) | (Post(p_i, t_j) = 0) | (p_i \geq 1))$$

Question 2.4 (1 point) *Donnez la définition d'une sous-classe de systèmes de transition à contrainte pour laquelle la conversion vers un réseau de Petri est directe.*

Il faut décrire la sous-classe correspondante à la question précédente.
- un ensemble de variables $\vec{s}$ sur N,
- un ensemble d'événements E,
- un ensemble de transitions $G(\vec{s}) \xrightarrow{e} \vec{s} := \sigma(\vec{s})$ où :
  - $\sigma(\vec{s})$ est restreint aux opérations $\{s_i := s_i + 1, s_i := s_i, s_i := s_i - 1\}$.
  - $G(e_j) = \bigwedge_{i=1}^{i=m} (s_i \geq k)$ avec $k \in \{0, 1\}$

Les applications Pre et $Post$ sont construites ainsi :

$$\forall e_j \begin{cases} \textit{poscondition} & \textit{precondition} & & \textit{Pre} & \textit{Post} \\ s_i := s_i - 1 & & \Rightarrow & Pre(s_i, e_j) = 1 & Post(s_i, e_j) = 0 \\ s_i := s_i + 1 & & \Rightarrow & Pre(s_i, e_j) = 0 & Post(s_i, e_j) = 1 \\ s_i := s_i & (s_i = 0) \Rightarrow \neg G(e_j) & \Rightarrow & Pre(s_i, e_j) = 1 & Post(s_i, e_j) = 1 \\ s_i := s_i & \neg((s_i = 0) \Rightarrow \neg G(e_j)) & \Rightarrow & Pre(s_i, e_j) = 0 & Post(s_i, e_j) = 0 \end{cases}$$

Définition 5 (Un réseau de Petri généralisé) *est un réseau de Petri dans lequel les deux applications*

$$\begin{aligned} Pre : & \quad P \times T \to \quad N \\ Post : & \quad P \times T \to \quad N \end{aligned}$$

spécifient que le franchissement d'une transition peut consommer plusieurs jetons dans les places en entrée, et déposer plusieurs jetons dans les places en sortie.

Question 2.5 (1,5 point) *Indiquez les modifications (si nécessaire) à apporter à votre solution à la question 2.1, afin d'avoir en entrée un* **réseau de Petri généralisé** *donné sous la forme du triplet de matrice (W^-, W^+, M_0), et toujours en sortie un système de transition à contrainte équivalent.*

Les postconditions (garde $True$) restent suffisantes pour les réseaux purs. Le problème provient encore des couples (transition, place) impurs. Il suffit donc de généraliser les gardes de l'exercice précédent. La garde de la transition t_j devient donc ((négation impure) ou (n jetons)) pour chaque place.

$$\bigwedge_{i=1}^{i=m} ((Pre(p_i,t_j)=0)|(Post(p_i,t_j)=0)|(\mathbf{p_i} \geq \mathbf{Pre(p_i,t_j)}))$$

Question 2.6 (1,5 point) *Donnez la définition d'une sous-classe de systèmes de transition à contrainte pour laquelle la conversion vers un réseau de Petri généralisé est directe.*

Il faut décrire la sous-classe correspondante à la question précédente.
- un ensemble de variables $\vec{s}$ sur N,
- un ensemble d'événements E,
- un ensemble de transitions $G(\vec{s}) \xrightarrow{e} \vec{s} := \sigma(\vec{s})$ où :
  - $\sigma(\vec{s})$ est étendu aux opérations $\{s_i := s_i + c\}$, ou $c \in Z$
  - $G(e_j) = \bigwedge_{i=1}^{i=m}(s_i \geq k)$ avec $k \in N$

Les applications Pre et $Post$ sont alors construites ainsi :

$$\forall e_j \left\{ \begin{array}{ll} poscondition & precondition \\ s_i := s_i + c & \text{plus petit } k \text{ tel que } (s_i < k) \Rightarrow \neg G(e_j) \end{array} \right. \Rightarrow \left\{ \begin{array}{l} Pre(s_i,e_j) = max(k,-c) \\ Post(s_i,e_j) = max(k+c,0) \end{array} \right. \quad rdp$$

Définition 6 (Un réseau de Petri généralisé avec arcs inhibiteurs) *est un réseau de Petri généralisé auquel on ajoute l'application*

$$ArcInh: \quad P \times T \rightarrow \quad \{false, true\}$$

qui énumère les arcs inhibiteurs. Le franchissement d'une transition t_j est alors possible ssi, d'une part t_j est franchissable dans le réseau de Petri généralisé ; d'autre part si pour toutes les places telles que $ArcInh(p_i,t_j) = true$, le nombre de jeton est zéro.

Question 2.7 (1,5 point) *Indiquez les modifications (si nécessaire) à apporter à votre solution à la question 2.1, afin d'avoir en entrée un* **réseau de Petri généralisé avec arcs inhibiteurs** *donné sous la forme du quadruplet de matrice $(W^-, W^+, M_0, ArcInh)$, et toujours en sortie un système de transition à contrainte équivalent.*

Il faut ajouter dans la garde une contrainte pour chaque arc inhibuteur. La garde devient :

$$\bigwedge_{i=1}^{i=m} (((Pre(p_i,t_j)=0)|(Post(p_i,t_j)=0)|(p_i \geq Pre(p_i,t_j)))\&(\mathbf{ArcInh(p_i,t_j)} \Rightarrow (\mathbf{p_i = 0})))$$

Question 2.8 (1,5 point) *Donnez la définition d'une sous-classe de systèmes de transition à contrainte pour laquelle la conversion vers un réseau de Petri généralisé avec arcs inhibiteurs est directe.*

Il faut décrire la sous-classe correspondante à la question précédente.
- un ensemble de variables $\vec{s}$ sur N,
- un ensemble d'événements E,
- un ensemble de transitions $G(\vec{s}) \xrightarrow{e} \vec{s} := \sigma(\vec{s})$ où :
  - $\sigma(\vec{s})$ est étendu aux opérations $\{s_i := s_i + c\}$, ou $c \in Z$
  - $G(e_j) = \bigwedge_{i=1}^{i=m} G(e_j, s_i)$ avec $G(e_j,s_i) = \left\{ \begin{array}{l} (s_i \geq k) \\ ou \\ (s_i = 0) \\ ou \\ (s_i = 0)\&(s_i \geq k) \end{array} \right.$

Les applications Pre, $Post$ et $ArcInh$ sont alors construites ainsi :

$$\forall e_j \begin{cases} \textit{poscondition} & \textit{precondition} & & \textit{rdp} \\ s_i := s_i + c & G(e_j, s_i) = false & \Rightarrow & \begin{cases} Pre(s_i, e_j) = 1 \\ Post(s_i, e_j) = 0 \\ ArcInh(s_i, e_j) = true \end{cases} \\ s_i := s_i + c & \exists k(\neg G(e_j, k_1) \& G(e_j, k_2)) \text{ pour } k_1 < k \leq k_2 & \Rightarrow & \begin{cases} Pre(s_i, e_j) = max(k, -c) \\ Post(s_i, e_j) = max(k + c, 0) \\ ArcInh(s_i, e_j) = false \end{cases} \\ s_i := s_i + c & (G(e_j, 0) \& \neg G(e_j, k)) \text{ pour } k > 0 & \Rightarrow & \begin{cases} Pre(s_i, e_j) = max(0, -c) \\ Post(s_i, e_j) = max(c, 0) \\ ArcInh(s_i, e_j) = true \end{cases} \end{cases}$$

Définition 7 (Un réseau de Petri généralisé avec arcs testeurs) *est un réseau de Petri généralisé auquel on ajoute l'application*

$$ArcTest : \quad P \times N \times T \rightarrow \quad \{true, false\}$$

qui décrit les arcs testeurs. Le franchissement d'une transition t_j est alors possible ssi, d'une part t_j est franchissable dans le réseau de Petri généralisé ; d'autre part pour le marquage courant M, pour toutes les places p_i, on a $ArcTest(p_i, M(p_i), t_j) = true$.

Question 2.9 (1,5 point) *Indiquez les modifications (si nécessaire) à apporter à votre solution à la question 2.1, afin d'avoir en entrée un* **réseau de Petri généralisé avec arcs testeurs** *donné sous la forme du quadruplet $(W^-, W^+, M_0, ArcTest)$, et toujours en sortie un système de transition à contrainte équivalent.*

Il faut remplacer dans la garde la contrainte sur les arcs inhibuteurs par celle sur les arcs testeurs. La garde devient :

$$\bigwedge_{i=1}^{i=m} (((Pre(p_i, t_j) = 0) | (Post(p_i, t_j) = 0) | (p_i \geq Pre(p_i, t_j))) \& \mathbf{ArcTest(p_i, p_i, t_j)})$$

Dans la notation $ArcTest(p_i, p_i, t_j)$, le premier p_i représente la variable et le second p_i représente sa valeur entière.

Question 2.10 (1,5 point) *Donnez la définition d'une sous-classe de systèmes de transition à contrainte pour laquelle la conversion vers un réseau de Petri généralisé avec arcs testeurs est directe.*

Il faut décrire la sous-classe correspondante à la question précédente.
- un ensemble de variables $\vec{s}$ sur N,
- un ensemble d'événements E,
- un ensemble de transitions $G(\vec{s}) \xrightarrow{e} \vec{s} := \sigma(\vec{s})$ où :
  - $\sigma(\vec{s})$ est étendu aux opérations $\{s_i := s_i + c\}$, ou $c \in Z$
  - $G(e_j) = \bigwedge_{i=1}^{i=m} G(e_j, s_i)$ avec $G(e_j, s_i) = ExpBool_j(s_i)$

Les applications Pre, $Post$ et $ArcTest$ sont alors construites ainsi :

$$\forall e_j \begin{cases} \textit{poscondition} & \textit{precondition} & & \textit{rdp} \\ s_i := s_i + c & ExpBool_j(s_i) & \Rightarrow & \begin{cases} Pre(s_i, e_j) = max(0, -c) \\ Post(s_i, e_j) = max(c, 0) \\ \forall n \in N ArcTest(s_i, n, e_j) = ExpBool_j(s_i)(n) \end{cases} \end{cases}$$

Question 2.11 (1 point) *Soit le système de transition à contrainte :*

```
variable   s : N;
evenement  a;
transition
  true |- a -> s++;
  true |- a -> s--;
```

Expliquez pourquoi vous ne pouvez pas traduire directement ce système vers un réseau de Petri ?

Le système de transition à contrainte est non déterministe, et les réseaux de Petri sont déterministes. La traduction ne peux donc pas être directe.

Question 2.12 (1 point) *Soit le système de transition à contrainte :*

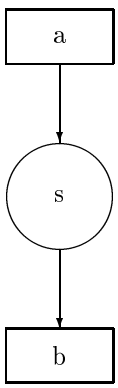
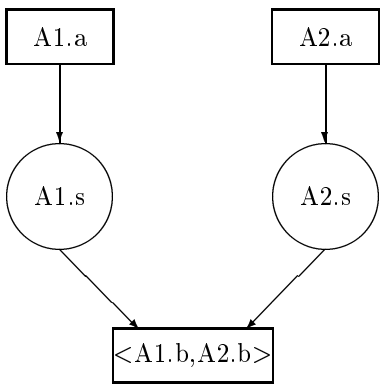
```
variable  s1, s2 : N;
evenement a;
transition
  s1<3 | s2 >5  |- a -> s1++, s2--;
```

Expliquez pourquoi vous ne pouvez pas traduire directement ce système vers un réseau de Petri ?

Les conditions de franchissement d'un réseau de Petri sont des conjonctions sur les places. Ici la garde est une disjonction entre deux variables. La traduction ne peux donc pas être directe.

Question 2.13 (1 point) *Soit le système de transition à contrainte :*

```
System A
variable  s : N;
evenement a, b;
transition :
  true  |- a -> s++;
  true  |- b -> s--;

System B
  sous-systeme A1, A2 de type A
  sync <A1.b, A2.b>;
```

Dessinez les réseaux de Petri des systèmes `A` *et* `B`.